TECHNISCHE
UNIVERSITÄT
DARMSTADT

Security in Mobile Peer-to-Peer Architectures – Introducing Mechanisms to Increase the Robustness of Overlay Routing Algorithms of Mobile Peer-to-Peer Architectures

Dem Fachbereich Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Dissertation

von

DIPL.-ING. CHRISTIAN GOTTRON

Geboren am 28. Oktober 1979 in Mainz, Deutschland

Vorsitz: Prof. Dr.-Ing Hans Eveking
Referent: Prof. Dr.-Ing. Ralf Steinmetz
Korreferent: Prof. Dr.-Ing. Michael Zink

Tag der Einreichung: 1. Februar 2013
Tag der Disputation: 22. März 2013

Hochschulkennziffer D17
Darmstadt 2013

ABSTRACT

A reliable communication platform is essential in disaster relief scenarios. Otherwise, an efficient coordination of the participating first response units cannot be ensured. However, during a disaster relief operation, a large amount of data is generated. Therefore, besides voice communication and the transmission of data such as text messages or pictures, storage and retrieval services are required. As a result, information such as medical data, a weather forecast, or data provided by other participating units can be stored in the network. Yet, recent large scale incidents like the terrorist attacks on the World Trade Center in 2001, the tsunami in the Indian Ocean in 2004, or small scale disasters like the explosions at the S.E. Fireworks company in 2000, have shown that centralized systems as cellular networks do not provide reliable services in such a scenario. This may be either a result of the damaged infrastructure due to the disaster or a result of high channel load, which has been observed after a disaster.

As a result, a more reliable communication platform is required, which satisfies the challenges introduced by a disaster relief scenario. This communication platform must not be based on a predefined infrastructure and needs to be able to handle the communication of mobile devices. Mobile Ad hoc networks are decentralized, mobile systems that are able to build a network on demand without a predefined infrastructure. Thus, those networks have been proposed by academia and industry to be used as communication substrate for such a disaster relief scenario. However, those Mobile Ad hoc Networks do not provide a storage and retrieval functionality. In this thesis, we propose to build a Mobile Peer-to-Peer system as a combination of a Mobile Ad hoc Network as underlay with a Peer-to-Peer overlay. While the Mobile Ad hoc network serves as a communication infrastructure, the Peer-to-Peer overlay provides the object storage and retrieval functionality. Yet, the underlay and the overlay have to be adapted to meet the challenges introduced by the disaster relief scenario in order to provide reliable and efficient services. To this end, Clustered Pastry, a new location aware Mobile Peer-to-Peer system is developed in this thesis.

The Clustered Pastry system inherits all the characteristics of the underlying architectures including multiple security issues. Therefore, an analysis of security threats in Mobile Peer-to-Peer scenarios is provided in the second part of this thesis. Known attacks against Mobile Ad hoc Networks and Peer-to-Peer networks are surveyed. Moreover, existing security mechanisms are discussed in the light of Mobile Peer-to-Peer scenarios in order to identify open security challenges. Based on those security challenges, new security mechanisms are developed for our Clustered Pastry system. Those mechanisms have to consider challenges introduced by the disaster relief scenario and by the characteristics of the Clustered Pastry system.

In summary, this thesis develops Clustered Pastry, a Mobile Peer-to-Peer system that can be deployed in disaster relief scenarios. Moreover, Clustered Pastry provides robustness to the attacks that have been identified as open security challenges. Thus, a reliable and secure storage and retrieval services can be provided by our new Clustered Pastry system.

KURZFASSUNG

Um eine effiziente Koordination der Einsatzkräfte in Katastrophenschutz-Szenarien zu ermöglichen ist eine zuverlässige Kommunikation essentiell. Neben der Sprachkommunikation und der Übertragung von Daten in Form von z.B. Textnachrichten oder Bildern, müssen weiterhin Daten im Netz gespeichert werden, die im Laufe einer solchen Operation entstehen. Hierbei kann es sich um z.B. medizinische Informationen, Wetterberichte, oder auch um Daten handeln, die von den Einsatzkräften vor Ort generiert wurden. Die Einsatzberichte diverser Katastrophen, wie der Anschlag auf das World Trade Center im Jahr 2001, der Tsunami im Indischen Ozean 2004 oder auch kleinere Katastrophen wie die Explosion der Feuerwerkskörperfabrik S.E. Fireworks in Enschede im Jahr 2000 haben gezeigt, dass oft die zentralisierte Kommunikationsinfrastruktur des Einsatzgebietes, wie zum Beispiel zelluläre Netze, nicht zuverlässig eingesetzt werden konnten. Dies resultierte entweder aus einer Beschädigung der Infrastruktur durch die Katastrophe selbst oder durch einen stark erhöhten Datenverkehr direkt nach der Katastrophe.

Als Folge dessen wird eine zuverlässigere Kommunikationsarchitektur benötigt, die den Herausforderungen eines solchen Katastrophenschutz-Szenarios genügt. Diese Kommunikationsarchitektur muss entsprechend den Einsatz mobiler Geräte unterstützen und unabhängig von einer bestehenden Infrastruktur operieren. Mobile Ad hoc Netze sind dezentrale Netze, die dynamisch eine Kommunikationsinfrastruktur zu generieren. Aufgrund dieser Eigenschaften wurden diese Netze in den letzten Jahren im Rahmen diverser Forschungs- und Industrieprojekte mit dem Schwerpunkt Kommunikation im Katastrophenschutz als potentielle Kommunikationsarchitektur vorgestellt. Allerdings bieten Mobilen Ad hoc Netze keine Dienste an die ein direktes Speichern und Verwalten von Daten ermöglichen. Entsprechend wird in dieser Arbeit ein mobiles Peer-to-Peer System entwickelt, welches jene mobilen Ad hoc Netze mit einem Peer-to-Peer Netz kombinieren. Hierbei wird das mobile Ad hoc Netz als Kommunikationsinfrastruktur verwendet, auf der das Peer-to-Peer Dienste zum Speichern von Daten anbietet. Durch die Kombination dieser beiden Systeme entstehen allerdings neue Herausforderungen die berücksichtigt werden müssen um ein zuverlässiges mobiles Peer-to-Peer System zu entwickeln. Aufbauend auf diesen Herrausforderungen wurde in dieser Arbeit das Clustered Pastry mobile Peer-to-Peer System entwickelt.

Clustered Pastry ist jedoch aufgrund der Charakteristiken der zugrundeliegenden Architekturen anfällig gegen diverse Angriffe. Entsprechend wird im zweiten Teil dieser Arbeit eine Sicherheitsanalyse des Clustered Pastry Systems durchgeführt. Dabei werden Schwachpunkte der zugrundeliegenden Systeme analysiert und mögliche Gegenmaßnahmen aus verwandten Arbeiten unter Berücksichtigung des Szenarios und der sich daraus ergebenden Herausforderungen diskutiert. Als Ergebnis werden offene Problemstellungen im Themenbereich der Sicherheit mobiler Peer-to-Peer Systeme identifiziert. Basierend auf den Ergebnissen dieser Analyse werden neue Sicherheitsmechanismen für das Clustered Pastry System entwickelt. Hierbei muss auf die Herausforderungen eingegangen werden die sowohl durch das Szenario

definiert werden als auch auf jene, die durch die Eigenschaften von Clustered Pastry entstehen.

Zusammenfassend wird in dieser Arbeit ein neues Mobiles Peer-to-Peer System entwickelt, welches den Anforderungen eines Katastrophenschutz Szenarios genügt. Darüber hinaus bietet das resultierende System Robustheit gegen ausgewählte kritische Angriffe, um einen zuverlässigen Einsatz zu ermöglichen. Als Ergebnis dieser Arbeit wird also ein zuverlässiges sicheres System vorgestellt, welches Daten dezentral speichern und abrufen kann.

ACKNOWLEDGMENTS

First of all, I would like to thank Prof. Ralf Steinmetz for his supervision and for the opportunity to work on my phd thesis at the Multimedia Communications Lab. I would further like to thank Prof. Michael Zink for the second assessment of my thesis and, moreover, for the chance to visit his department at the University of Massachusetts.

Moreover I owe sincere and earnest thankfulness to my colleagues at the Multimedia Communication Lab. Especially I want to thank Andrè König and Sonja Bergsträßer as they supported me since my very first day at the institute. I also want to thank my group leaders Andreas Reinhardt and Doreen Böhnstedt for their help during my last year as phd student. Moreover, I would further like to show my gratitude to Matthias Hollick as he provided invaluable feedback for my work.

I also want to thank the SOC group at KOM, especially Melanie Siebenhaar for her last minute feedback for my thesis and Andrè Miede for the LaTex template. Moreover, I want to thank those people that keep the institute running as Karola Schork-Jakobi, Silvia Rödelsperger, Jan Hansen, Gisela Scholz-Schmidt, Marion Ehlhardt, Frank Jöst, Sabine Kräh, Monika Jayme, and Christian Theiß.

Of course, I would not be able to finish this thesis without the support of my friends and family. Thus, I want to thank my farther Walter, my mother Monika, and my grandmother Marianne for their support as well as my brother Thomas, his wife Isabella, and their children Emily, Lara and Julius. Beside my own parents, I also want to show my gratitude to the parents and grandmother of my fiancée, to Anneli, Peter and Rita. I would also like to thank Anna, Nico, Laura, Piet, Daniel, Alex, Azze, Thomas, Katrin, and all of my other friends for ensuring that I get some off time of my thesis.

Last but not least, I want to thank my fiancée Katrin, for supporting me during the last four years and for her understanding regarding my increased workload on some weekends and holidays.

Darmstadt 2013 C. G.

In loving memory...

... August, Max, Leo, and Maria.

CONTENTS

1

INTRODUCTION

»In creating, the only hard thing is to begin:
a grass blade's no easier to make than an oak.«

— James Russell Lowell

1.1 MOTIVATION

Today, we live in a world where nearly all kind of information is readily available when requested. Therefore, we are used to retrieve, e.g., the weather forecast, news, or information about traffic jams on our way home whenever we require this information. Due to the technological progress of the last decade, this data can also be accessed in a mobile environment by using cellular networks. As a result, modern mobile phones have evolved from voice communication units to devices that are capable of exchanging multiple kinds of data.

Those cellular networks suffice to fulfill the challenges introduced by mundane scenarios. However, those networks may be considered as unreliable in disaster relief scenarios. As shown by recent incidents, cellular networks can be unavailable due to damage introduced by a disaster [57] or as a result of high channel load [60]. However, a reliable communication platform that provides the first responding units with services such as voice communication and that ensures an efficient distribution of information is vital in disaster relief scenarios [27].

Due to these facts, several research projects have been conducted recently in order to develop a reliable decentralized communication platform to ensure that first responding units are able to coordinate themselves efficiently. Wireless, mobile networks based on a decentralized paradigm called Mobile Ad hoc Network (MANET) are a promising approach to provide a reliable communication infrastructure in disaster relief scenarios. MANETs were harnessed by multiple projects as a solution for this challenge [57]. Besides their application in academia, MANETs were further advertised by commercial projects like HiMoNN[1] in order to ensure a reliable communication of first responding units.

Although MANETs were proposed initially in the late 90's of the previous century, it is only in the last decade that first real world applications have been deployed as part of DUMBO [57], the one laptop per child project[2], and the German Freifunk[3] initiative. MANETs provide a communication platform and enable a data transmission between distant nodes. Yet, applications are required that operate on this platform in order to provide services. This includes the storage and retrieval of data objects such as text messages or pictures. As information may be vital in a disaster relief scenario [109], the storage and distribution of data is essential to provide the first response units data on causalities, the disaster site (e.g., status reports), or sensor information

1 http://himonn.iabg.de/
2 http://one.laptop.org/
3 http://start.freifunk.net/

(e.g., smoke detectors alerts). Therefore, Mobile Peer-to-Peer (MP2P) systems have been introduced recently. Those systems combine a Peer-to-Peer (P2P) overlay with a MANET underlay in order to ensure a completely decentralized storage of data in the network.

In the last decade, MP2P systems have been developed for a large set of scenarios. This includes military [105], vehicular [66] and disaster relief settings. However, when considering disaster relief scenarios, the proposed MP2P systems have been mostly developed based on harnessing notebooks or PDAs and do not consider opportunities that have been provided by the new generation of mobile devices such as smart phones. For example, multiple MP2P systems harness underlay mechanisms in order to roughly estimate the geographical location of a node while most of today's mobile devices provide more precise mechanisms such as the Global Positioning System (GPS). Other MP2P systems still require centralized entities in order to operate efficiently. However, an MP2P system can benefit from using today's technologies.

Moreover, security challenges in the context of MP2P systems have mostly been neglected by now. The few security mechanisms that have been developed for *MP2P* systems ensure, e.g., the privacy of the participants [43] or the authentication of devices that join the network [112] [26]. Yet, the availability of the services provided by an MP2P system is still an open security challenge. Even though, considering the disaster relief scenario, robustness against maliciously behaving or faulty nodes is vital in order to ensure the availability of the communication platform.

In summary, most existing MP2P systems that have been designed for disaster relief scenarios do not consider opportunities provided by the new generation of mobile devices. Thus, those architectures may provide better results when harnessing technologies such as GPS. Moreover, security threats have mostly been neglected in the context of MP2P scenarios. Yet, reliable and secure services are vital in the context of disaster relief scenarios.

1.2 GOALS

The major objective of this thesis is to develop an MP2P system that harnesses state-of-the-art technology like GPS to provide efficient and reliable storage and retrieval services. Moreover, this system has to be robust against multiple types of attacks. As a result, the following goals are addressed in this thesis.

- Developing a decentralized MP2P system that provides storage and retrieval operations of small data items in a disaster relief scenario where no infrastructure is available. We consider only disaster relief scenarios in this thesis, even though the resulting architecture may also be used in the context of other scenarios.
- Analyzing security threats for MP2P due to maliciously behaving participants and, in particular, routing attacks.
- Developing security mechanisms for MP2P system based on the results of the survey of security threats.

1.3 RESEARCH CHALLENGES

The characteristics of an MP2P system satisfies the basic requirements introduced by the disaster relief scenarios as mentioned in the motivation of this thesis. Yet, those decentralized systems also introduce a new set of research challenges. Those are mostly a result of the fusion of the underlying network types and are based on their characteristics. These research challenges have to be addressed in order to ensure the reliable usage of MP2P systems in disaster relief scenarios.

WIRELESS UNDERLAY

Most P2P systems were developed in the context of a reliable underlay such as the Internet. Therefore, the design of those systems was based on an underlay that provides a high bandwidth, a mostly stable topology, and introduces a low fraction of lost packets. Most of the P2P system algorithms such as the routing tables were developed based on these assumptions. Yet, when introducing a mobile, wireless underlay, those assumptions are not valid anymore. On the wireless channel, packets may get lost as a result of collisions and, furthermore, the bandwidth is strongly limited.

DECENTRALIZED SYSTEM

Both, P2P systems and MANETs are decentralized networks. This characteristic is required to satisfy the challenges introduced by the disaster relief scenario. Moreover, this characteristic also introduces new challenges for the resulting MP2P system. As the MP2P system is completely decentralized, challenges arise that affect the basic functionality of the network and as well the robustness of this system.

MOBILITY OF PARTICIPANTS

Due to the disaster relief scenario, mobility of the participants has to be assumed. As a result of the mobility of the devices, a dynamic topology has to be considered. This affects the availability of the services provided by the P2P algorithm directly. Furthermore, also data stored on nodes that leave the network due to mobility may affect the availability of the provided services.

COOPERATION OF PARTICIPANTS

As no predefined infrastructure is available, routing functionality provided by participating nodes is required whenever a message has to be to delivered to its destination. However, maliciously behaving nodes may exploit this fact in order to deny the services provided by the MP2P system.

1.4 CONTRIBUTIONS

This thesis introduces a new location-aware MP2P system that has been developed for disaster relief scenarios. This system is, moreover, robust against security threats, which are introduced by the P2P lookup mechanism and the allocation of data objects.

The contribution of this thesis is organized in three major parts: The first part focuses on developing an MP2P system suited for disaster relief scenarios. The second part focuses on analyzing open security threats for this system. The last part provides

security mechanisms that are developed in this thesis to meet the identified security challenges.

DEVELOPING A LOCATION AWARE MOBILE PEER-TO-PEER SYSTEM

Clustered Pastry as a location-aware MP2P system is the first contribution of this thesis. This system combines a MANET underlay with a Distributed Hashtable (DHT) overlay to provide storage and retrieval services in a mobile setting as required by our disaster relief scenario. Moreover, we have to adapted these underlying systems to meet the challenges that arise due to a wireless, mobile underlay. Additionally, multiple mechanisms have been developed in order to ensure reliability and efficiency of this MP2P system, including overlay routing tables that are updated based on information gathered by the underlay and mechanisms that reduce the traffic generated by the overlay.

ANALYSIS OF SECURITY THREATS OF MOBILE PEER-TO-PEER SYSTEMS

MP2P inherits the vulnerabilities to a large set of attacks from their underlying systems. Therefore, security threats and, in particular, routing attacks for MP2P are discussed and analyzed as part of this thesis. Furthermore, three routing attacks have been discussed more precisely, as those attacks are capable to affect the availability of the services provided by the MP2P system.

DEVELOPMENT OF SECURITY MECHANISMS FOR MOBILE PEER-TO-PEER SYSTEMS

Security mechanisms for each of the identified threats will be developed for the Clustered Pastry system in order to ensure robustness against the previously mentioned routing attacks. These mechanisms will be developed in the light of the challenges and requirements that are introduced by an MP2P system.

- The first mechanism is an adapted replication mechanism that provides copies of each object in order to ensure the availability of the services of the network. This mechanism harnesses the highly structured characteristics of the overlay and combines them with the location awareness of the underlay in order to ensure an efficient distribution of the replicas.
- MP2P systems have to rely on benign behavior of other participants in order to be able to locate a content provider. Thus, maliciously behaving nodes may exploit this characteristic of the network and deny those lookup services. Therefore, a security mechanism will be developed to monitor the lookup mechanism of the MP2P system.
- In order to retrieve a data object in an MP2P system, the address of the content provider has to be determined. Yet, a maliciously behaving node may exploit the requesting node's unawareness about the address of the content provider and may pretend to be the content provider in order to deny this object. Therefore, a mechanism will be developed to validate a content provider.

1.5 THESIS STRUCTURE

This thesis is structured in seven chapters:

After this first chapter, which introduces and motivates the topic of this thesis, foundations and related work is surveyed in Chapter 2. Thus, information on MP2P systems is provided as background information in the first part of the second chapter. This includes an overview of the underlying architectures. Furthermore, MANET and P2P systems are discussed in detail that are being used as a substrate to develop our clustered MP2P system. Moreover, related work on MP2P is surveyed in the chapter. Challenges of MP2P security and existing security mechanisms are discussed in the second half of Chapter 2. Therefore, selected attacks on the underlying systems are discussed. Existing mechanisms, preceded by related work, that ensure robustness against those attacks, are surveyed as well.

In Chapter 3 we develop our Clustered Pastry system. In a first step, the challenges introduced by the disaster relief scenario are discussed. Thereafter, the basic concepts of the MP2P system are developed based on those challenges. During the course of the chapter implementation details are provided. This includes the structure and update functionality of the routing tables as well as mechanisms to increase the network's robustness and reduce the generated network traffic. The chapter concludes with a comparison of Clustered Pastry against two MP2P systems that have been introduced in related work, which share some characteristics with our system.

The resulting Clustered Pastry system is evaluated with regard to the performance in disaster relief scenarios in Chapter 4. Therefore, different scenario settings are introduced that are used to estimate the efficiency of Clustered Pastry. Those scenarios are based on parameter settings, such as the number of nodes in the network, the mobility model, or background traffic. Furthermore, recommended default values for the parameters of our system are determined during this evaluation.

Chapter 5 discusses security threats for MP2P systems. Therefore, threats introduced by the underlying systems are analyzed. Based on existing work in the area of network security in MANET, P2P, and MP2P security, open challenges are identified. As a result of this discussion, challenging attacks on the MP2P system are analyzed and the impact of those attacks is evaluated at the end of the chapter.

Based on the security challenges analyzed in the Chapter 5, security mechanisms are developed in Chapter 6. Therefore, challenges introduced by the characteristics of MP2P system are analyzed in order to develop mechanisms that are adapted to the needs of the scenario. At the end of the chapter, those mechanisms are evaluated and compared to security mechanisms proposed by related work.

The last chapter summarizes the content and outcomes of this thesis. Furthermore, the contributions of this thesis are compared to the goals introduced in this chapter. The last chapter concludes with a discussion of future work that may be based on this thesis.

2

FOUNDATIONS AND RELATED WORK

»Learn from yesterday, live for today, hope for tomorrow.
The important thing is not to stop questioning.«

— Albert Einstein

The goal of this chapter is to introduce the foundations as well as security threats and solutions for MP2P systems. Furthermore, scientific work is reviewed that is closely related to the topic of this thesis.

In the first part of this chapter, the general structure and the underlying architectures of MP2P systems are introduced. Therefore, foundations of P2P and MANET protocols are revisited in this chapter. Furthermore, related work in the field of MP2P systems is reviewed. The second part of this chapter focuses on security threats that affect the routing mechanisms of MP2P systems. Three prominent and harmful attacks are introduced that affect the routing mechanism and the services provided by the system. Furthermore, security mechanisms proposed by related work are surveyed that increase the robustness of MP2P networks against those attacks.

2.1 NETWORK AND COMMUNICATION SYSTEMS

As a first step in developing a novel MP2P system, the underlying protocols and the structure of the resulting MP2P network are discussed. In this section, models are introduced that are used to design this system. Moreover, as MP2P systems combine MANET with P2P networks, prominent existing protocols are surveyed in the following subsections. Furthermore, related work on MP2P systems is introduced at the end of this section.

2.1.1 *Communication Models and Concepts*

Network models are used to describe the structure of a network and the interaction of the underlying protocols. Those models are used to reduce the complexity of the resulting system. In this thesis two basic models are used: The layer model and the concept of overlay networks.

The layer model classifies protocols based on their functionality. Thus, we use this model to develop the new MP2P system and to define the functionality of the deployed protocols. The concept of overlay networks in terms of building a network on top of another network is used in multiple architectures including P2P systems. Therefore, the overlay paradigm is also discussed in this section.

LAYER MODELS

Today, most of the digital devices such as desktop computers, smart phones, notebooks, or even MP3 players are able to communicate with other devices. Therefore,

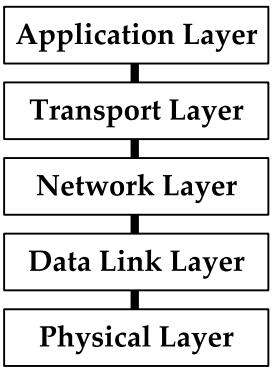


Figure 1: Tanenbaum's layer model

a vast amount of protocols are available and required to, e.g., handle connections, retransmit packets, discover routes to the destination node, negotiate the transmission rate, and so on. In order to reduce the complexity of the resulting protocol structure of a node, layer models are used. Those models consist of a stack of layers, where each layer provides specific services to the layer above. Therefore, protocols can be assigned according to their functionality to a layer (or, in some cases, to multiple layers). The topmost layer of the model is used as interface to the user and the lowest layer is connected to the network. Every layer but the lowest layer communicates with adjacent layers only. The lowest layer also may use the channel to communicate with other nodes. In this thesis, the layer model of Tanenbaum [107] is used. This model combines the Open Systems Interconnection (OSI) model [2] of the International Organization for Standardization with the TCP/IP model [15] of the Internet Engineering Task Force. As shown in Figure 1, Tanenbaum's model consists of 5 layers. The lowest layer, the physical layer, handles the transmission of bits over the channel. The data link layer manages the communication with adjacent nodes. This includes flow control and framing of the transmitted data as well as error detection and correction. Furthermore, the data link layer maintains the medium access control and, therefore, determines when a node is allowed to send data over the channel. Often, e.g., when transmitting data via the Internet, the destination node is not directly connected to the source node, but the data has to be forwarded by multiple intermediate nodes or routers. Therefore, routing services provided by the network layer protocols are required in order to discover a route to the destination node. The fourth layer, the transport layer establishes and provides end-to-end communication services for applications and further services like congestion control. The last layer, the application layer includes multiple kinds of applications, e.g., electronic mail services.

In some cases, information may be required at a particular layer, that can not be provided by the surrounding but another layer. Therefore, cross-layer optimization is used to provide feedback from layers beyond the virtual borders of the model.

OVERLAY NETWORKS

Overlay networks add an addition layer of indirection to the layer model. Therefore, the overlay is introduced as a virtual network that is built on top of an underlying network, henceforth called the underlay. While the underlay provides the connectivity between the participants of the networks, virtual links are harnessed by the overlay.

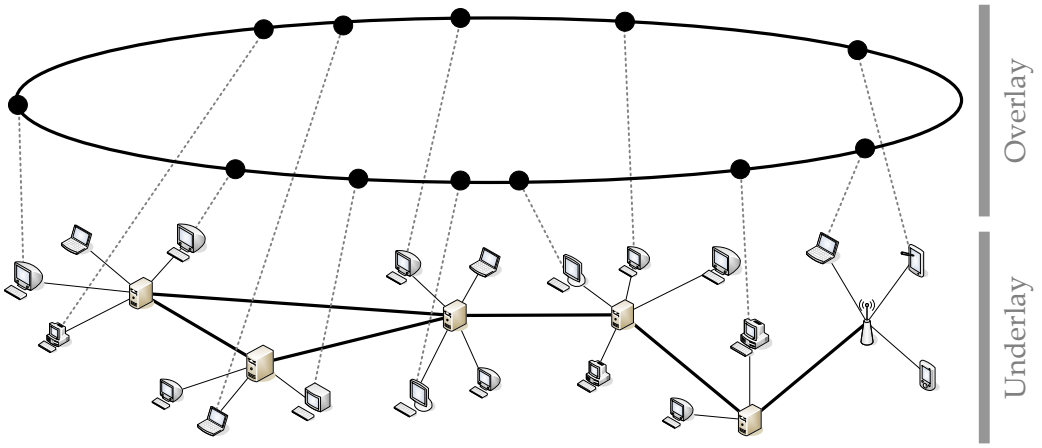


Figure 2: Example for an overlay network

P2P networks are a prominent example for overlay networks. The P2P overlay provides services, such as decentralized storage and retrieval of data objects, but requires an underlay in order to establish routes between the participating peers. In most cases, the Internet is used as the underlying network. However, not every node in the underlying network has to participate in the overlay as well, as shown in the example in Figure 2. Theoretically, a P2P overlay network can run on top of any underlay that provides the required services.

2.1.2 *Mobile Ad hoc Networks*

MANETs are decentralized networks established spontaneously by nodes that communicate via a wireless channel. As the wireless transmission range of those nodes is limited by the transmission power, the destination of a message may not be within the direct transmission range of the sender. Therefore, intermediate nodes are required that have to provide routing functionality on demand to forward a message to the destination. The distance between the sender and the destination in MANETs is often defined by the number of times a message is forwarded. This distance metric is also called the hop distance. As a result of this definition, multi-hop communication is assumed whenever a message has to be forwarded by at least a single intermediate node.

As a result of this decentralized and distributed characteristics of these networks, single points of failure can be avoided. Routing tasks of a node that leaves the network unexpected, e.g., due to technical issues, can be performed by other nearby nodes. Due to these characteristics, MANETs do not depend on a previously defined infrastructure. Furthermore, in most scenarios a major part of the network nodes are mobile. This results in a highly dynamic topology. Moreover, the bandwidth is strongly limited due to the wireless channel shared with every node within transmission range. Thus, adapted routing algorithms are required to address these challenges.

CLASSIFICATION OF PROTOCOLS FOR MOBILE AD HOC NETWORKS

Several routing protocols for MANETs have been proposed in the recent years. These protocols operate mostly on the network layer and are able to establish routes to

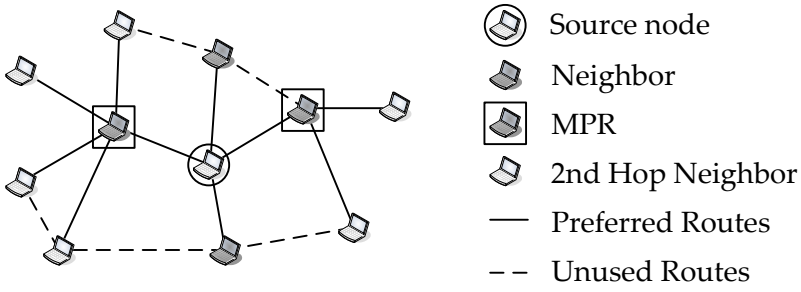


Figure 3: Multi Point Relays selected by a node that uses the Optimized Link State Routing protocol

nodes in the network. In this thesis, the MANET protocols are categorized according to their routing characteristics into *proactive*, *reactive*, and *hybrid* approaches.

Proactive protocols maintain routes to every node in the network. As a result, valid routes are readily available when required. Yet, periodic updates of the routing tables are required, as otherwise the dynamic topology will result in stale routing table entries. This results in traffic overhead that is generated by these update messages. The Optimized Link State Routing protocol (OLSR) [21] is an often referenced example of a *proactive* MANET routing protocol. Because of the special importance of the OLSR protocol in this thesis, we will introduced this protocol in detail a few paragraphs later.

In contrast to *proactive* approaches, *reactive* algorithms initiate a routing process on demand only. As the topology in MANETs may change constantly due to node mobility, routing overhead may be significantly reduced when a route discovery is initiated only when a route is required. On the downside, establishing a route on demand delays the data transmission. The Ad hoc On-Demand Distance Vector (AODV) [80] and Dynamic Source Routing (DSR) [56] routing protocols are well-known and often referenced *reactive* protocols for MANETs.

As a combination of *reactive* and *proactive* protocols, *hybrid* protocols combine the characteristics of both. Routes to nodes that are within a specific distance to the source are maintained proactively while routes to distant nodes are established on demand only. The Zone Routing Protocol [41] is an example for a *hybrid* MANET protocol.

OPTIMIZED LINK STATE ROUTING PROTOCOL

The OLSR protocol is one of the few MANET algorithms that has not been evaluated solely in the lab via simulation or an analytical approach. Instead, it has been validated in multiple real world testbeds, e.g., in the DUMBO [57] or the Freifunk [1] project. As mentioned before, OLSR is a *proactive* MANET routing protocol and, therefore, updates the routing tables of each node periodically.

This protocol detects geographical neighbors by sending periodically *hello* messages. Due to these messages, information regarding geographical neighbors can be obtained and used to update the local routing table. Furthermore, information about the detected neighbors is distributed by Topology Control (TC) messages in the network. In order to reduce overhead generated by those routing table update messages, TC messages are forwarded by selected nodes only. Therefore, each node defines a set of geographical neighbors. This set of selected neighbors, called Multi Point Relay (MPR),

has to provide a connection to every second hop neighbor of the node as shown in Figure 3.

OLSR relies on a hop-based metric to estimate the distances between nodes and to derive the routing tables. However, testbed evaluations of hop-based metrics have shown that they are inefficient [70][38]. Therefore, the Expected Transmission Count (ETX) [23] extension of OLSR was introduced. The routing path metric that is used by the ETX extension also considers the maximum throughput of a link besides the overall hop distance. Thus, routes established with the ETX metric are more reliable.

2.1.3 *Peer-to-Peer Systems*

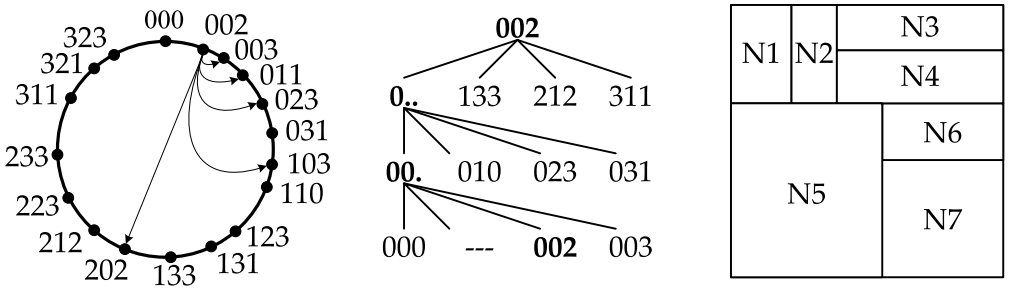
As mentioned before, P2P systems are overlay networks mostly based on the Internet as underlying network. Due to the decentralized architecture, P2P systems are robust against single node failures. Whenever a node leaves the network (either intentionally or due to node failure), another peer is able to provide the leaving node's functionality. In contrast to the traditional centralized Client-Server architectures, the overall load is, on average, equally distributed among all peers in the network. Yet, new challenges arise due to the decentralized characteristics of P2P networks, for example, the dynamic participation of peers in the network (churn) [104]. As each peer provides content, these objects have to be redistributed when a content provider leaves the network.

In recent years, multiple P2P architectures were introduced. Those can be categorized into *unstructured* and *structured* architectures [103]. In the following paragraphs, the characteristics of those architectures are discussed. Furthermore, a prominent and efficient P2P system called Pastry is introduced at the end of this subsection.

UNSTRUCTURED PEER-TO-PEER ARCHITECTURES

The *unstructured* architectures can be subdivided in *centralized*, *pure*, and *hybrid* P2P architectures. The first generation of P2P systems were *centralized* P2P architectures. They still required a central entity that coordinates the lookup functionality. Due to this fact, those systems have some features and drawbacks of a Client-Server based architecture. Every peer that wants to share content in terms of data objects with other nodes in the network has to notify the central entity. Whenever another peer requests this object, a lookup request is sent to this central entity. As the routing table of the central entity maps content that is stored within the network to the address of the content provider, the central entity responds to the request with the address of the peer that has stored the requested object. As a result, the routing and maintenance costs of the peers in the network are very low as compared to other P2P architectures. Yet, the central entity has to provide a high amount of resources, as the routing functionality has to be maintained and managed by this node. Furthermore, this entity is a single point of failure as the lookup process relies on the centrally maintained routing table. The most prominent example of a *centralized* P2P system is Napster [103].

In order to provide completely decentralized services, *pure* P2P systems were developed. Those systems do not require a central entity. Thus, *pure* P2P networks have no single point of failure and each peer has to provide the same services and



(a) *Ring* structure (Chord) (b) *Tree* structure (Tapestry) (c) *Hypercube* structure (CAN)

Figure 4: Structure of Peer-to-Peer architectures

is, therefore, treated equally. The lookup mechanisms of *pure* P2P systems are based on flooding lookup requests. Therefore, lookups can be resolved even though the address of the node that is responsible for the requested content is unknown. However, the traffic overhead generated by the storage and retrieval of an object is increased strongly due to the flooding based routing mechanism. Freenet [20] and Gnutella 0.4 [61] are examples of *pure* P2P architectures.

The third kind of *unstructured* P2P architectures are the *hybrid* P2P architectures. *Hybrid* P2P systems combine the characteristics of both, *pure* and *centralized* P2P systems. Some few nodes (super-peers) providing a high amount of resources, are used to reduce the routing overhead. Every peer that joins the network is assigned to such a super-peer. This super-peer is responsible for the node and has to store information regarding the node's address and the content provided by this node. During routing, requests are not flooded through the whole network, but only sent to the super-peers. As a result, this subset of nodes is able to provide information regarding the nodes and the content stored in the network. The amount of requests that have to be sent is reduced to the number of super-peers. Thus, the traffic is strongly reduced compared to a *pure* P2P system. Examples for *hybrid* P2P architectures are Gnutella 0.6 [61], KaZaA [59], or the location aware Globase.KOM [65].

STRUCTURED PEER-TO-PEER ARCHITECTURES

Structured P2P architectures (DHT) are completely decentralized. But in contrast to *pure* P2P, no costly flooding-based lookup mechanism is used. Instead, DHTs harness unique overlay identifiers to identify peers as well as objects that are stored in the network. Due to these overlay identifiers, the virtual distance between two nodes or an object and a node can be derived. The routing algorithms harness this virtual distance between node identifiers during a lookup. The routing tables of the nodes are highly structured and provide addresses of a well-defined sparse set of the nodes and the according node identifiers only.

Lookups are used to resolve the overlay identifier to the Internet Protocol (IP) address of a node. In most cases, only the destination identifier but not the network address is known by the source. Due to this fact, the node has to check the routing table for the identifier of the destination. If the address of the destination is found, the lookup can be completed within a single overlay hop. Otherwise a request message

is sent to a node with an identifier that is logically closer to the identifier of the destination. This intermediate node has to check its own routing table to determine a next hop node according to the routing algorithm to forward the message. Due to this fact, the message is forwarded towards the destination identifier until the message is received by the destination node. This destination node replies a response message to the source to complete this lookup.

When a node provides content to the network, the object identifier of this new object has to be generated in the first step (e.g., by hashing the object's name). Thereafter, a lookup for this object identifier is initiated. The reply message that has been received as a result of this lookup includes the address of the node that is responsible for the new object (root node). This root node either stores the new data object or a link to the content provider. To retrieve an object, a node also has to initiate a lookup. The resulting reply message provides the address of the root node of the requested object. Thereafter, the source is able to download the requested object either from the root node or the content provider directly.

Several DHTs have been developed in recent years. Those systems can be classified according to the structure of their routing table. In the context of this thesis, three basic structures are discussed. These are the *ring*, *tree*, and *hypercube* structure.

Ring based architectures distribute the node identifiers on a circular identifier-space. Starting with the identifier that has the lowest numerical value on the top position, the identifier is increased clockwise. Chord [102] is a prominent example for a *ring* structured DHT. In Figure 4a, an example of the structure of a Chord routing table is shown.

Tree structured architectures, like for example Tapestry [121], maintain a prefix-based routing table. As shown in Figure 4b, each row i of a routing table provides links to nodes with i matching prefix digits to the identifier of the owner of this routing table. Whenever a lookup request is initiated or received by a node, the identifier of the destination is compared with the node identifier. The length of the matching prefix determines the row of the routing table where the address of the next hop node is stored. Thus, both, the routing table size as well as the average required number of hops during a lookup scales logarithmic to the network size.

Hypercube structured DHTs like the Content Addressable Network (CAN) [85] are based on a d-dimensional identifier-space. Nodes are represented by an area within this identifier-space as shown in Figure 4c. Objects are identified by a singular point in the identifier-space and assigned to the node that is responsible for this area.

Furthermore, some P2P systems combine the algorithms of two architectures. Pastry [88] for example is based on differently structured routing tables. One is *tree*-based and the other *ring* structured.

PASTRY

Many MP2P systems (including the system that has been developed as part of this thesis) are based on the Pastry DHT. Therefore, the structure and algorithms of this P2P system is surveyed in the following paragraphs in detail.

The Pastry DHT, as introduced by Rowstron et al., is based on a *hybrid* architecture. This P2P system has been developed for large scale scenarios with static nodes that are connected via a wired underlay, e.g., the Internet. In order to provide an efficient storage and retrieval service in these scenarios, the number of routing table entires and the number of hops between source and destination are a logarithmic function of

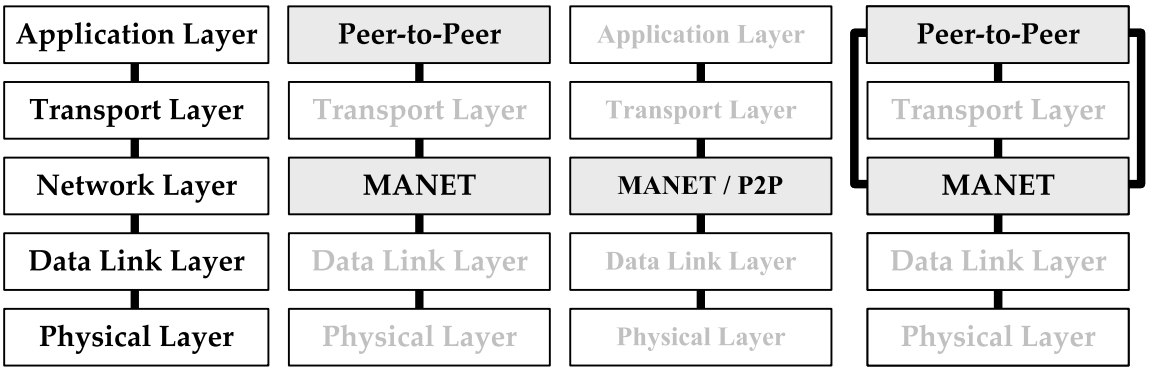
the number of overall nodes in the network. Each node and each object is identified by a unique 128-bit overlay identifier. This identifer is used to define the virtual position of the node in the overlay network. As a result, the routing algorithm is based on the identifiers and their virtual location in the identifier-space. These identifiers are represented by a sequence of digits that is defined by the parameter b. Each digit is displayed as a value between 0 and $2^b - 1$. By default, the parameter b is set to 4 and, therefore, each digit is mapped to a hexadecimal value. Three routing tables are maintained by Pastry. Each routing table maps the identifiers of a specific set of nodes to the proper IP addresses.

The first routing table of Pastry is *tree* structured and provides links to nodes that are logically distant. The functionality of this *routing table* is similar to Tapestry's routing table (as shown in Figure 4b). Each row contains $2^b - 1$ entries. The n^{th} row provides links to nodes that share $n - 1$ digits in the prefix and differ in the n^{th} digit. The parameter b defines how many entries are stored in the *routing table* per row. As multiple nodes satisfy the requirement of each entry, a proximity metric is used to determine the geographically closest node. Therefore, a ping message is sent to each node that satisfies the requirements. Thereafter, the node with the lowest round trip time of the ping message is stored at the *routing table*. The *routing table* is not maintained proactively but on demand only. Whenever a node sends a message and does not receive a response, the node, the message has been sent to, is assumed as unavailable. Thereafter, this link is removed from the table and an update request is sent to another node that is stored in the same row as the unavailable node. In the case that no other link to a node is available in this row or no node within this row is able to provide a link to a matching node, the request is sent to any other node that is stored in the *routing table*.

The second routing table, the *leaf set*, stores the addresses of the virtual neighbors. This routing table is *ring* structured. The *leaf set* is defined to a size of L nodes (per default $L = 2^b$). Half of the entries provide links to nodes with a numerically smaller identifier and the other half provides links to nodes with numerically larger identifier compared to the identifier of the *leaf set* owner. Similar to the *routing table* the *leaf set* maintains the routing table entires on demand only. Whenever a leaf node does not respond to a message, this node is assumed to be offline. If a stale entry is detected, the virtual neighbors are notified. These neighbors reply with a message containing their *leaf set*. Based on those *leaf sets*, the own local *leaf set* can be updated.

The *neighborhood set* is the third routing table maintained by Pastry. This table stores links to nodes that are geographically close to the owner of the routing table. Therefore, the proximity metric is used to determine the distance between the nodes. Yet, this routing table is not used during the routing but can be used, e.g., to provide a geographically close bootstrapping node when disconnected from the P2P network (for further details on bootstrapping see also the last paragraph of this subsection).

In order to route a request to the destination, each node that receives this request has to check the *leaf set* first. The *leaf set* can be harnessed to determine whether the node itself or a virtual neighbor is the destination of this request. If so, a reply message is sent to the source of the request or the request is forwarded to the destination node, respectively. Otherwise the *routing table* is used to to determine a node that matches the identifier of the destination by at least one more digit. Thereafter, the message is forwarded to this node that is logically closer to the destination. The average number



(a) Basic layer model (b) *Layered* model (c) *Integrated* model (d) *Cross-layered* model

Figure 5: Mobile Peer-to-Peer structures

of hops (h) is logarithmic in the parameter b and the number of nodes in the network (N) (see Function 2.1).

$$h = \log_{2^b}(N) \tag{2.1}$$

In order to join the network, the address of a Pastry node that participates in the P2P network is required (the bootstrapping node). After generating the identifier of the new node, a join message is routed via the bootstrapping node to the virtual neighbors. The addresses of intermediate nodes that forward the join message are used to fill the *routing table*. After the join request has arrived at the virtual neighbor, the *leaf set* table is generated based on information provided by this neighbor.

2.1.4 *Mobile Peer-to-Peer Systems*

After discussing the foundations of the underlying networks, foundations on MP2P architectures are provided in this section. Therefore, a classification of MP2P systems is introduced and, furthermore, challenges are discussed. Based on this classification, related work in the area of MP2P systems is discussed.

Several systems that combine MANET with P2P networks have been introduced recently. Those systems can be classified according to the implementation of the P2P algorithm into *layered*, *integrated* and *cross-layered* approaches as shown in Figure 5.

In the following, each of the approaches is described and related approaches are discussed. Further details on the classification of MP2P systems can be found in our survey [37].

LAYERED MOBILE PEER-TO-PEER SYSTEMS

Layered MP2P architectures simply combine existing MANET with P2P systems. As a result, underlay and overlay are strictly separated and, therefore, no direct communication between the two underlying protocols is enabled, as shown in Figure 5b. However, most *layered* MP2P systems adapt the underlay and/or the overlay system to meet the challenges that arise due to the combination of these architecture. In particular, the traffic overhead that is generated by the overlay is reduced, due to the limited bandwidth of the MANET underlay.

Castro et al. proposed an MP2P system that is based on the Bamboo DHT [18]. The routing algorithm of this DHT is very similar to Pastry's routing algorithm, but differs in the maintenance of the routing tables. As mentioned in Chapter 2.1.3, Pastry updates the routing tables on demand only. However, Bamboo harnesses periodic updates to provide fresh routing table entries. Castro et al. reduces the traffic that is generated by the overlay by increasing the update intervals. Thus, this MP2P system is able to operate despite the limited bandwidth of the MANET underlay.

Moreover, location awareness is used by several MP2P systems that have been proposed in the last decade. Therefore, the geographical position of a node is considered during the lookup. Thus, the traffic that is generated due to a lookup is reduced. The PeerNet [34] MP2P system maps the identifier space of the overlay to specified zones in the deployment area. Moreover, the mobility profile and history of the nodes is used to map these nodes on these zones in order to define virtual residences. Highly available proxies are required at these zones that provide information about the location of the nodes that reside in this virtual area. Therefore, each node that leaves its virtual residence has to notify the proxy and has to provide a mobility profile. This mobility profile includes information that can be used to estimate the destination of this node. Based on this mobility profile and due to the location aware underlay routing mechanism, PeerNet is able to establish routes in a efficient way.

Millar et al. [74] proposed a landmarking-based, location aware MP2P system. The landmarking system splits the namespace of the overlay identifier in equal sized zones and defines a landmark node at each zone. This node periodically sends landmarking messages. These messages are forwarded by each node within the landmarking zone. Thus, nodes that are located at the boarder between two landmarking zones receive the landmarking messages of both zones. Based on the hop count of the received landmarking messages, those nodes decide to which landmarking zone they belong. All nodes that are located in the same landmarking zone share the same prefix of the overlay identifier. By changing the landmarking zones, those nodes have to adapt their overlay identifier. As a result, virtual neighbors are located in the same geographical area. Thus, a location aware routing is enabled, which results in a reduced overhead during the routing. Moreover, Millar et al. use a Bamboo DHT as overlay and, therefore, reduce the maintenance overhead by reducing the update intervals.

INTEGRATED MOBILE PEER-TO-PEER SYSTEMS

Integrated MP2P systems combine the MANET and the P2P protocols at a single layer as shown in Figure 5c. Thus, traffic overhead can be reduced as no separate routing tables for the MANET and P2P system are required. Moreover, information provided by the MANET protocol can be used by the P2P system.

The Scalable Source Routing [29] is an *integrated* MP2P system that is based on the Chord DHT. Thus, the Scalable Source Routing system is *ring* structured. Yet, only links to direct virtual and geographical neighbors are provided by the routing tables of the MP2P system. Thus, whenever a lookup request is received, this request is either forwarded to a geographical neighbor that is virtually closer to the destination or to a virtual neighbor. This results in a strongly reduced traffic overhead that is generated by the maintenance of the routing tables.

EKTA [84] and DPSR [52] are both based on the Pastry and integrate this DHT at the network layer. Furthermore, both systems harnesses forwarded or overheard

messages to update the routing tables. Thus, the maintenance overhead in terms of traffic can be reduced. Both systems maintain a *leaf set* that provides multi-hop routes to the virtual neighbors. Those routes are updated whenever a message to such a node is forwarded or overheard.

CROSS LAYERED MOBILE PEER-TO-PEER SYSTEMS

A *cross-layered* approach is a combines the benefits of the *layered* and *integrated* architectures. Due to cross-layering, information provided by the network layer can be used by the P2P system to, e.g., reduce traffic generated during the routing table maintenance. Moreover, the underlying architectures do not have to be restructured but only minor adaptations of the MANET and P2P protocols are required.

The *cross-layered* MADPastry [118] is based on the Pastry DHT and uses a similar landmarking mechanism as proposed by Millar et al. [74]. Therefore, location aware routing algorithm is harnessed in order to reduce the traffic generated by a lookup. Moreover, the routing tables of MADPastry are truncated. Thus, the *routing table* is used for the first overlay hop only. Thereafter, the *leaf set* is used to route the lookup request. Moreover, both routing tables are maintained by cross-layer information obtained from overheard messages. Thus, traffic overhead that is generated by the routing table maintenance is strongly reduced.

CHALLENGES OF MOBILE PEER-TO-PEER SYSTEMS

Both P2P and MANET systems share central characteristics. Combining them results in a fully decentralized architecture that provides storage and retrieval functionality in a mobile environment. However, MP2P systems introduce major challenges due to the limited resources provided by the MANET underlay. As most DHTs were developed in the context of a static underlay like the Internet, overlay systems have to be adapted in order to match the requirements introduced by the mobile, wireless underlay. Therefore, one challenge is the overhead generated by the overlay that has to be reduced. This includes traffic due to routing table maintenance, the lookup and the bootstrapping mechanism.

2.1.5 *Summary of Related Work in Mobile Peer-to-Peer Systems*

In this section, foundations of communication models, MANET and P2P systems have been provided. Furthermore, related work on MP2P systems as a combination of MANET and P2P systems have been surveyed. MP2P systems are decentralized mobile systems that inherit beneficial but also challenging characteristics of the underlying architectures. Due to this fact, MANETs can not be simply combined with a P2P overlay but adaptations are required to address these challenges. In the recent years, multiple MP2P systems have been introduced. These systems can be classified according to their implementation method of the overlay. *Layered* MP2P systems as discussed in Section 2.1.4 allows only limited adaptations to meet the challenges of MP2P systems. *Integrated* systems on the other hand require significant modifications on the underlying architectures. Thus, it is not possible to adapt an *integrated* approach to be used in combination with another underlay or overlay protocol. However, *cross-layered* systems provide the benefits of *integrated* system but can be easily deployed in combination with different protocols. Therefore, the

cross-layered architecture is assumed as the most flexible and efficient structure for an MP2P system and is used to develop a novel MP2P system in this thesis.

2.2 SECURITY IN MOBILE PEER-TO-PEER SYSTEMS

Decentralized systems as MANET, P2P, or MP2P systems are vulnerable to a wide range of malicious behavior. Therefore, security concerns have to be considered in order to ensure that the system provides reliable and efficient services.

This section provides foundations of security goals. Those goals introduce basic requirements that have to be fulfilled by every communication system in order to ensure a secure and reliable communication [94] [25]. Security mechanisms proposed by the related work as well as remaining security threats on MP2P systems are discussed subsequently. As will be shown, routing attacks on the MP2P overlay have been mostly neglected by now. Those attacks are able to affect the reliability of the services provided by the system strongly and will thus be focus of the second part of this thesis. However, even though only few related work regarding security and robustness of the routing mechanism of MP2P systems exists, several security mechanisms have been proposed for the underlying architectures. Hence, security mechanisms were also surveyed that increase the network's robustness against routing attacks in the field of MANET and P2P systems.

2.2.1 *Security Goals and Basic Definitions*

General security requirements are often represented by three basic goals [94] known as the CIA-triad (*Confidentiality*, *Integrity*, *Availability*). Those goals have to be satisfied by every communication system in order to ensure a secure and reliable functionality. Thus, these requirements are also relevant in the context of MP2P systems.

- **Confidentiality:** The security goal of *confidentiality* is satisfied if only those nodes are able to access data or to use services that are authorized to perform this task.
- **Integrity:** The requirements introduced by the *integrity* are satisfied when neither data can be modified undetected nor unauthorized. Due to this fact, tampered data can be detected. Furthermore, digital rights management is based on data integrity.
- **Availability:** According to Schneier [94], »availability is about ensuring that an attacker can't prevent legitimate users from having reasonable access to their system«. Though, the definition of »reasonable access« depends on the specific scenario and application.

Besides these basic security goals, further goals have been introduced by, e.g., Eckert [25]. The extended security goals provide more precise requirements and cover further challenges that have not been considered by the CIA-triad as follows.

- **Authentication:** Each entity and each object has to be able to *authenticate* itself by using a unique identifier. Due to this fact, a node or object can verify its own claimed identity whenever requested (e.g., when a communication is initiated).

- **Non Repudiation:** An entity should not be able to deny an operation previously performed. This extended security goal is of particular importance in the area of e-commerce.
- **Privacy:** Unauthorized entities should not be able to retrieve information that could be used to identify, to observe, or to spy out an user of another entity. This security goal does not ensure the reliable functionality of the system, but a reasonable handling of the private data of users.

Beside the security goals, definitions have been established to describe a system's vulnerabilities, the probability and impact of malicious behavior, and the malicious behavior itself. Thus, security *threats*, *risks*, and *attacks* have been defined by Eckert [25] as follows.

- **Threat:** A *threat* is the result of the systems vulnerability that can be exploited to affect the availability, confidentiality or integrity of the systems services.
- **Risk:** The *risk* is defined by the probability that a system is affected by malicious behavior and the resulting effect on the system.
- **Attack:** Maliciously behaving participants may attack the system in order to compromise the network. As a result, the attacker may access services or the network itself without permission. Furthermore, data or services may be modified, deleted, denied, or information about the other participants may be extracted by means of this attack.

2.2.2 *Survey of existing Mobile Peer-to-Peer Security Mechanisms*

Even though several MP2P architectures have been developed in recent years, only few mechanisms have been proposed to ensure the security and reliability of those networks. In the following paragraphs, existing security mechanisms are surveyed that have been developed in the context of MP2P scenarios.

The goal of the EU-funded PEPERS project [113] was to develop a platform to support the design of secure MP2P applications. Therefore, a methodology was introduced that identifies the weakness of a mobile application according to its structure. Furthermore, mechanisms like a monitoring system that is able to detect policy violation on the application layer were discussed within the scope of this project [100]. However, the outcomes of this project can rather be seen as a guide that provides hints on how to develop a secure MP2P application. Specific mechanisms that increase the networks resilience against malicious behavior have not been developed.

Basic authentication mechanisms for MP2P have further been introduced. Čapkun et al. [112] developed two different access control mechanisms for MP2P architectures. In the first scenario, authentication via a offline certification authority was assumed. For this reason, each node in the network is aware of the public keys of the other participants. The second scenario was based on a decentralized approach. Each node has to be authenticated via a face to face authentication. Thus, the authentication is performed by the user of this device directly. Another approach introduced by Fenkam et al. [26] uses a decentralized access control mechanism. Furthermore, an admission control mechanism has been introduced by Manulis [71], that considers privacy issues of the joining nodes.

Beside privacy during admission control, privacy considerations during lookups have been discussed in the related work. Han and Liu [43] introduced an algorithm that ensures privacy for both the initiator and the responder of a lookup. In contrary to approaches that have been proposed in the context of P2P, this approach is not based on costly onion routing [33] but on Shamir's Secret Sharing [97]. Based on this approach, Tsai et al. introduced an improved approach that also ensures data integrity protection [108].

Adapted replication mechanisms proposed by Mondal et al. [75] [76] are based on super-peers which are assumed to have high battery power and processing capacity. Those nodes may not move freely but have to stay within a specified geographical area. All other nodes have to report future movement plans and required objects to the super-peer as they assume that each node has a predefined schedule anyway.

In summary, few security mechanisms have been designed for MP2P systems recently. Yet, routing attacks have been mostly neglected by the related work. Solely, the proposed replication mechanisms can be used to improve the networks robustness against those attacks. However, as routing attacks strongly affect the availability of the services provided by the MP2P system, replication mechanisms alone do not suffice to ensure reliable and robust services.

2.2.3 *Incorrect Lookup Routing Attack*

Routing attacks have a high impact on the availability of the services provided by the MP2P system. Even a limited number of malicious nodes may be able to deny a high fraction of the network's services. In the following paragraphs, the *Incorrect Lookup Routing Attack* as introduced by [99] is revisited. This attack exploits the routing algorithm of the overlay to attack the network and is, therefore, able to affect the availability of the networks services.

In order to lookup an object or a node, DHTs have to rely on the benign behavior of intermediate nodes. Those nodes have to forward the request message towards the destination. Yet, a benign behavior of those intermediate nodes can not always be assumed. The *Incorrect Lookup Routing Attack* exploits this vulnerability of the DHTs routing algorithm. Malicious nodes discard or redirect incoming route requests instead of forwarding them. As a result, the routing mechanism of DHTs and, therefore, the *availability* of objects stored in the network are affected by the *Incorrect Lookup Routing Attack*.

Castro et al. [17] introduced an analytical model that describes the effects of malicious nodes that drop packets on the efficiency of a recursive lookup mechanism. The packet delivery ratio (σ) is based on the average number of hops (h) per request and the fraction of malicious nodes (f) (as shown in Equation 2.2).

$$\sigma = (1 - f)^h \tag{2.2}$$

For example, considering a setting composed of 100.000 nodes that are participating in a Pastry DHT. The number of average hops (h) is a function of the overall number of participants as discussed in Section 2.1.3 (see also Equation 2.1). As a result 4.15 hops are required on average to complete a lookup in this scenario. Furthermore, we assume that 10% of the particpants behaves maliciously (f). As a result of the

parameters of this setting, less than 65% of the lookups initiated can be completed successfully according to Equation 2.2.

ROBUSTNESS AGAINST INCORRECT LOOKUP ROUTING ATTACK

The *Incorrect Lookup Routing Attack* strongly affects the routing algorithms and, therefore, the *availability* of the services provided by the DHT. Due to this fact, several security mechanisms were proposed to increase the network robustness against this attack.

By enabling the source node of a lookup to determine the route of the request message directly, malicious behavior of intermediate nodes can be bypassed. Sit and Morris [99] proposed an *iterative routing mechanism* that is based on this concept. Whenever a request message is received by a node, this message must not be forwarded to a node closer to the destination, but the node has to reply to the source of this lookup. This reply message includes a set of next hop addresses that can be used as next hop nodes. Thus, the source node is able to determine an adequate next hop node and to direct the request to this node. As a result, intermediate nodes that discard or redirect a request message can be detected easily. Whenever no reply message or a reply message with a faulty set of addresses (according to the routing algorithm) is received, malicious behavior can be assumed. Thereafter, the source node is able to forward the request message to any other node that has been provided by the previous reply message in order to bypass the malicious node. On the downside, overhead is generated due to the reply messages sent by intermediate nodes. Furthermore, a transmission delay is introduced when the network is under attack, as the source node has to wait for the reply messages for a specific amount of time. In addition, overhead is generated as request messages have to be resent whenever a malicious node drops a request. Several other security mechanisms have been introduced in the recent years that are based on a similar approach. For example, Myrmic [114] and Sechord [77] harness the source node of a lookup to coordinate the routing of the request message and to detect misbehavior of intermediate nodes during the lookup.

Another approach to improve the network robustness is to introduce redundancy directly to the DHT's routing mechanism. To this end, the *redundant routing mechanism* has been proposed by Castro et al. [17]. When a lookup is initiated, the redundant mechanism does not send a single request message, but sends multiple messages in parallel over multiple routes. Due to this mechanism, the probability is increased that at least a single request is received by the destination, yet overhead in terms of an increased number of request messages per lookup is generated. In order to improve the redundant routing algorithm, Srivatsa and Liu [101] analyzed the influence of independent routing paths on the efficiency of this approach. As a result, disjoint routes have been identified as an essential requirement for the reliability of this approach. Otherwise, a single malicious node might be able to drop or misroute every request in order to deny the lookup functionality. However, in most DHTs independent routing paths cannot be guaranteed without a high effort. Other approaches, such as, e.g., HALO [58] or Cyclone [6] are also based on a redundant routing mechanism.

Hildrum and Kubiatowicz [51] combined the redundant with the iterative routing algorithm. As a result, a *parallel and iterative routing mechanism* has been introduced. Each request is sent multiple times on parallel routes. Furthermore, instead of forwarding the message directly, intermediate nodes have to send reply messages to

the source. Those messages including a set of next hop nodes. The combination of the redundant with the iterative routing algorithm results in an increased robustness against the *Incorrect Lookup Routing Attack*. Yet, the drawbacks regarding overhead and delay of both underlying algorithms are inherited by this approach.

Reputation-based mechanisms have been developed to reduce the impact of the *Incorrect Lookup Routing Attack*. Sánchez-Artigas et al. [91] combined the redundant routing algorithm with a reputation system. The resulting resistant routing mechanism sends a request message to the next hop node as well as messages to each of the n neighbors of this node. Each of these nodes forwards the request again to a node logically closer to destination node and to the n neighbors of that node. Due to this fact, the request message is not only received by the destination node but also by the n neighbors of this node. By storing replicas at those neighbors, this algorithm provides a set of nodes that includes each root of the n replicas as well as the root of the original object. The reputation system is used in order to reduce the overhead. Each node in the network monitors the behavior of the virtual neighbors. Whenever a node is detected as malicious node, this node is neglected during subsequent lookups. Furthermore, the routing has to be coordinated as otherwise the traffic may be strongly increased when neighbors forward a request to different nodes (e.g., as a result of stale routing table entries). Therefore, an iterative approach can be used to enable the source to coordinate the routing. However, as each node has to forward $n + 1$ request messages at each step of the routing, a high overhead is generated. Further approaches harness reputation of routes and nodes to determine the most reliable as routing path, e.g., the *Higher-Reputated Neighbor Selection* [92], the *Exclusion Routing Protocol* [87]. Another routing mechanism harnesses social links to provide more reliable lookups [73].

RELATED TOPICS IN THE AREA OF MOBILE AD HOC NETWORKS

Also MANETs have to rely on intermediate nodes during routing and during data transmission. Thus, similar routing attacks such as the *Incorrect Lookup Routing Attack* can be performed on those networks. As a result, multiple security mechanisms have been introduced in recent years that improve the robustness of P2P systems against routing and forwarding attacks.

A very promising approach has been introduced by Marti et al. [72]. The *WatchDog* Intrusion Detection System (IDS) is able to detect routing attacks in MANETs. Assuming bidirectional wireless links, each node in a wireless network receives every message sent by their geographical neighbors. WatchDog harnesses those overheard messages to detect misbehavior. After a node A has sent a message to an intermediate node B, this node has to forward the message towards destination. However, a malicious node may either modify or drop the message. By using this intrusion detection system, the sender of the message (node A) has to monitor each message that is sent by the intermediate node B. Due to this fact, both cases of misbehavior can be detected by node A. Whenever either no message has been forwarded within a specific amount of time or the message differs from the initially sent message, a malicious behavior can be assumed. Yet, when the message has been forwarded correctly to node C, node B has to monitor node C unless this node is the destination of the message.

Marti et al. proposed to use information regarding detected misbehaving nodes during routing. Due to this reputation based routing mechanism, malicious nodes

should be avoided. WatchDog has been evaluated in a testbed by Buchegger et al. [16]. As a result, Buchegger et al. were able to show that WatchDog can be used to detect maliciously behaving nodes in a real wold scenario. However, WatchDog has been developed for MANETs only and therefore provides a reliable detection of malicious nodes within the underlay. As the mechanism is based on IP address solely, no detection of malicious behavior in the overlay can be provided.

2.2.4 *Forged Lookup Replies*

Besides dropping lookup requests, reply messages can also be forged by malicious intermediate nodes. When a lookup for an object is initiated, the source of this lookup is unaware of the root node's overlay identifier. Castro et al. [17] introduced an attack that exploits the unawareness of the source about the virtual distance between object and root node. Therefore, each malicious node that is numerically closer to the overlay identifier of the object than the source node itself is able to claim to be the root. As a result, the malicious node is able forge a reply message. This message may either include a faulty object or may claim that the requested object or service is not available in the network. This affects the *availability* of the services and objects provided by the system.

MECHANISMS FOR DESTINATION VALIDATION

Some few mechanisms have been proposed in the recent years that are able to detect forged reply messages in P2P scenarios. The most promising approaches are surveyed in the following paragraphs.

In order to validate received reply messages, Castro et al. [17] proposed the *Routing Failure Test*. Castro assumes that the node identifiers are equally distributed in the identifier space. Therefore, and due to the overall low number of maliciously behaving nodes, the *Routing Failure Test* is able to detect a forged reply message. This promising approach is discussed in detail in Chapter 6.5.

Wang et al. introduced the DHT routing mechanism Myrmic [114]. Myrmic harness certificates to certify the identifier-space, a node maintains. Due to those certificates, forged lookup messages can be simply detected. Yet, in order to provide those certificates, a Neighborhood Authority is required. Whenever this entity is not available, new nodes cannot join the network.

An other approach introduced by Ganesh and Zaho [30] stores certificates of nodes that are participating in the DHT at so called proof managers. Whenever malicious behavior is assumed, the proof manager of the determined destination node is contacted in order to receive the information whether or not the node has behaved maliciously. This approach introduces overhead whenever a node either joins the network and certifies itself at the according proof manager and whenever a malicious behavior is assumed.

2.2.5 *Storage and Retrieval Attack*

Even after a successfully completed routing process, a lookup may still fail due to a maliciously behaving root node. This node may perform a *Storage and Retrieval Attack* [99] and deny the access to the requested object. As it is hard to distinguish between

unavailable and denied objects during a lookup, most P2P systems are not inherently able to detect this attack. The efficiency of this attack depends directly on the fraction of malicious nodes, since in the general case objects are equally distributed on the nodes in the network.

The impact of the *Storage and Retrieval Attack* can be strongly reduced by distributing copies of each object (replicas) in the network. As a result of the object replication, an object can only be denied when the root of the object as well as each root of every replica behaves maliciously. However, several approaches were introduced in recent years providing replicas to MANET and DHT networks. In the following, the most prominent approaches for replicas are introduced.

REPLICA DISTRIBUTION IN PEER-TO-PEER SYSTEMS

Replication mechanisms developed for DHTs are highly structured but mostly unaware of the geographical location of the distributed replicas. Basic mechanisms as proposed by Castro et al. [17] or Rowstron et al. [89] store replicas at the virtual neighbors of the root node. This is further used by several previously mentioned mechanisms, e.g., Sánchez-Artigas et al.'s resistant routing. Other replication mechanisms distribute the replicas within the identifier space according to a specific algorithm. Harvesf and Blough [48] [49] proposed to distribute the replicas equally in the overlay. This results in disjoint routing paths for each of the replicas in, e.g., *tree*-based architectures. A similar approach has further been introduced by Ghodsi et al. [32]. Also few location aware replication mechanisms such as replica enumeration [10] were proposed, which distributes replicas in the network. Whenever an object has to be downloaded, probes are sent in order to determine the replica that is located geographically close to the requesting node.

REPLICA DISTRIBUTION IN MOBILE AD HOC NETWORKS

Replication mechanisms developed for MANETs often harness the structure of the underlay to distribute replicas. Therefore, a basic awareness of the geographical location of replicas is harnessed to optimize the replication mechanism. Hara [44] proposed a MANET replication mechanism that harness the access frequency of objects to allocate a replicas and to determine the number of required replicas. This approach has been introduced in order to prevent data loss due to network partitioning and to improve the network performance. The mechanism that has been initially proposed by Hara neglected the update functionality of the replicas. Therefore, Hara proposed a static update function in [45] and a dynamic update function in [46]. Furthermore, Hara et al. [47] optimized the metrics by including the correlation of the objects. In order to ensure that a replica in a MANET is within a specific distance to each node in the network, Tamori et al. [106] proposed a replica distribution scheme which is based on a fixed distance between two replicas of the same object. The distance is defined by the hop count. A similar approach was proposed by [55]. Most of the replication mechanisms developed for MANETs are unstructured and often require a flooding based mechanism to distribute new objects. Yet, some few replication schemes are controlled at the application layer, e.g., the mechanism proposed by Bellavista et al. [12] [13]. This approach uses a replication manager that was selected in a decentralized way, which coordinates the replication mechanism. Yet, overhead is introduced due to the selection of a manager and the management of replicas.

2.2.6 *Summary Related Work in Mobile Peer-to-Peer Security*

In this section, basic definitions and goals in the area of network security have been provided. These foundations can be regarded as universally valid and, therefore, have to be considered in MP2P scenarios.

Few mechanisms have already been proposed to increase an MP2P system's robustness against a wide range of different attacks. Yet, routing attacks that affect the *availability* of objects and services have not been investigated by now. However, a high *availability* is an essential requirement for MP2P systems, especially when considering scenarios such as disaster relief. Therefore, three prominent routing attacks have been discussed in this section that are able to negatively affect the networks functionality strongly.

The *Incorrect Lookup Routing Attack* has been introduced as major threat to the lookup mechanism. As a result of this attack, request messages are discarded or redirected. The second attack introduced in this chapter is based on forging reply messages. As a result, the source node either receives a faulty object or no object at all. The last attack, discussed in this section is based on a malicious root node. This root node denies the requested object. Moreover, security mechanisms proposed for P2P systems and MANETs have been surveyed.

However, it is unclear whether these mechanisms satisfy the requirements of an MP2P system. Therefore, these mechanisms have to be reviewed in the light of MP2P scenarios. If the findings of this review indicate that the existing mechanisms do not suffice to ensure a robust MP2P system, novel security mechanisms are required that are adapted to the challenges introduced by the MP2P scenario.

3

A CLUSTERED MOBILE PEER-TO-PEER ARCHITECTURE

»I begin with an idea and then it becomes something else.«

— Pablo Picasso

Nowadays, most mobile communication devices, e.g., smart phones, are able to communicate with each other via an 802.11 Wireless Local Area Networks (WLAN) standard. Furthermore, those devices are generally able to locate their geographical position by using the Global Positioning System (GPS). Due to this technical progress, location-aware decentralized mobile communication architectures are enabled.

MP2P systems are based on these technologies. These systems operate in a decentralized manner and do not have to rely on a predefined infrastructure. Due to these characteristics and as they enable a mobile, wireless communication, those systems may be used in wide range of application scenarios. This includes challenging scenarios as disaster relief, where the previously mentioned characteristics are highly required. The MP2P system can be used as a communication platform and is able to provide storage and retrieval services.

In the previous chapter, related work and basic foundations have been discussed. The results have shown that even though multiple MP2P systems have been developed in recent years, only few harness location awareness. Furthermore, existing approaches either require central entities to manage node mobility or are based on a landmarking system (see also Chapter 2.1.4).

In this chapter, Clustered Pastry an clustered, location aware MP2P system is introduced that has been developed in the context of this thesis [39]. First, challenges that have to be considered in a disaster relief scenario are introduced in the first part of this chapter. Thereafter, the underlying concepts and the resulting structure of our system are discussed in detail. Based on these details, the algorithms and components of the MP2P system are defined. Once, those basic components have been selected, the implementation details of the previously introduced concept are described extensively. This chapter concludes with a summary of our Clustered Pastry's system's structure and a discussion of the benefits of this system.

3.1 SCENARIO

The first step in designing our Cluster Pastry system is to outline the application scenario and the resulting challenges. The findings of this analysis can be used to develop a novel concept that is adapted to the needs and requirements of a mobile, wireless, and decentralized MP2P system.

3.1.1 *Disaster Relief Scenario*

In this thesis we focuses on communication in disaster relief scenarios. A disaster is defined by the International Strategy for Disaster Reduction (ISDR) of the United Nations as follows:

> »A serious disruption of the functioning of a community or a society involving widespread human, material, economic or environmental losses and impacts, which exceeds the ability of the affected community or society to cope using its own resources.«
>
> — ISDR [54]

As disasters »exceed the ability of the affected community or society to cope using its own resources«, external help in terms of emergency services is required. These emergency services include first responding units.

As the coordination of these units at a disaster site is essential, a reliable communication platform is required [109]. This includes, beyond voice communication, the distribution and availability of information [22] as data about the affected area, maps [27] or medical issues [19]. Furthermore, the amount of data, which is gathered during a relief operation, increases vastly [27]. Therefore, a reliable storage mechanism is required in order to ensure the availability of this information.

In most disaster relief scenarios, multiple organizations such as, firefighters, medical units or the police have to cooperate[109]. As a result of this situation, a communication beyond the boundaries of these different agencies is required [9]. Furthermore, a mission report of a recent large scale disaster [22] indicates that an easy to deploy on demand communication platform may be beneficial to ensure a reliable communication already in the first hours of the relief operation.

Due to the scenario, we have to assume that at least a part of the participating units are mobile [8]. Therefore, the communication system has to cope with mobile devices. Moreover, when considering a worst case scenario, a disaster may result in a strongly damaged disaster site. Therefore, the infrastructure (including the communication infrastructure) of this affected area may be damaged or completely destroyed. In any case, the functionality of the wireless infrastructure based systems as cellular networks cannot be assumed [60]. As the first responding units have to operate at the disaster site, they may be affected as well by rough environment that has resulted due to the disaster. As a result, the communication devices may get damaged. Thus, single points of failure must be avoided as otherwise the communication platform may be unusable when only a few devices get damaged.

Due to this scenario, a system is required that satisfies multiple requirements. This includes mobile, robust, decentralized, and infrastructureless characteristics. Moreover, this system has to be available on demand.

3.2 CONCEPT OF OUR MOBILE PEER-TO-PEER SYSTEM

These requirements can be met by a combination of a MANET underlay and a P2P overlay. Both architectures are completely decentralized and are, therefore, able to operate on demand as they are not based on a predefined infrastructure. Furthermore, both systems are able to provide functionality even when single node fails. By combining those two systems, an MP2P system is built. This system inherits the

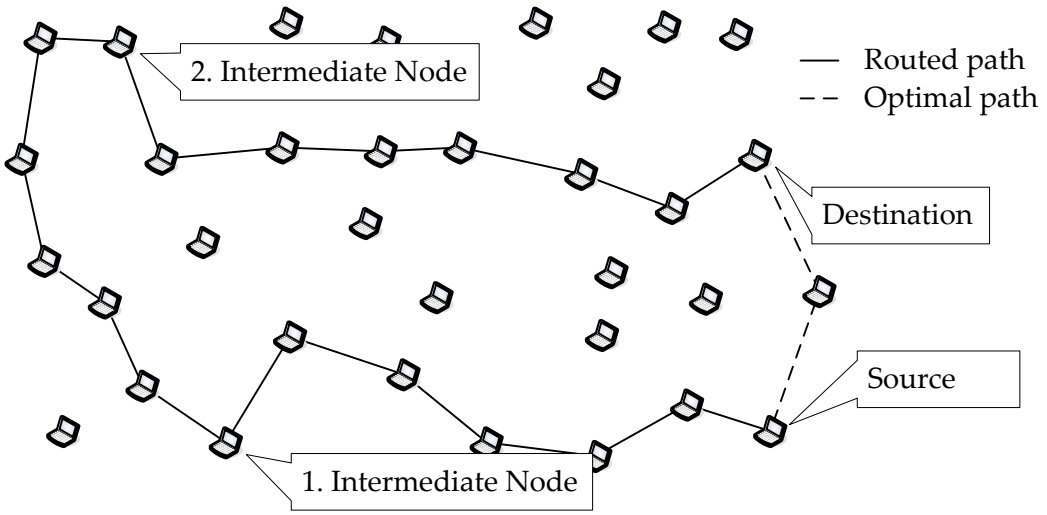


Figure 6: Example of a routing in system with randomly distributed overlay identifier

benefits of both underlying systems and, due to the MANET underlay, is further completely mobile. In summary, an MP2P system suffice the requirements introduced by the previously discussed disaster relief scenario.

However, the MANET underlay and the P2P overlay were not developed to be used as a combined system. Thus, new challenges arise and, therefore, the resulting system has to be adapted in order to operate efficiently. These challenges as well as an adapted design concept to overcome those challenges are discussed in the rest of this section.

3.2.1 *Challenges*

Traditional P2P systems are based on a network of static nodes such as desktop PCs that provide, e.g., file-sharing or text and voice-chat functionality. In most cases a large amount of data like video files or audio streams are distributed among peers in those networks. Furthermore, a large number of participants is assumed (more than 10,000 nodes) that are distributed around the world and are connected via the Internet. Due to these characteristics, the goals of static P2P systems are focused on providing decentralized services, load balancing, network robustness, and enabling network scalability.

However, most MP2P architectures were developed in the context of completely different scenarios. When combined with a MANET, the number of participants is strongly limited due to the characteristics of the underlay. Instead of hundreds of thousand nodes, only a few hundred nodes are feasible due to bandwidth limitations and the overhead generated as a result of the multi-hop communication. However, scenarios considered in this thesis meet those limitations. First response scenarios are limited in most cases to approximately a hundred [7] in medium scale and only rarely up to a thousand participants in large scale scenarios such as the terrorist attacks on the World Trade Center in New York [60]. Yet, not every unit is equipped with a communication device. Furthermore, the deployment area is rather geographically small compared to traditional P2P networks, which connect nodes around the world and

provides services as file sharing [83] or video streaming [122]. As the first responding teams operate within an area that is predefined by the disaster site, the deployment area can be assumed as static. In addition, we have to assume mobile nodes such as smartphones or notebooks instead of static desktop computers. However, the node mobility depends on the specific characteristics of the scenario. While the average node speed is high in vehicular networks, first responding scenarios introduce a rather low node speed (when neglecting vehicles). Furthermore, we assume that it is more likely that text files such as status reports or medical information or small pictures of causalities are stored within an MP2P network due to the characteristics of the scenario. As a result, the goals of MP2P architectures do not focus on scalability issues or load balancing but rather on the availability of stored objects. However, we assume that network robustness is even more important in the context of MP2P systems as, due to the mobility and the wireless channel, network partitioning may occur.

Furthermore, security challenges arise due to the combination of the underlying systems. Those have to be solved in order to ensure that the networks services can be provided in a reliable way. However, this topic will be discussed in depth in the ensuing chapters.

3.2.2 *Design Concept*

As mentioned in Chapter 2.1.3 DHTs use unique identifiers to address nodes within the overlay. Those identifiers are randomly generated. Thus, it can be ensured that the identifiers of the nodes are distributed uniformly in the identifier space. This provides on one hand robustness against lookup and retrieval attacks. The node identifier also defines which objects a node has to maintain. When a node is able to choose its own identifier, this node would also be able to choose the objects it is responsible for. As a result, an object can be denied by a malicious node with little effort. On the other hand, load balancing in the network is assured due to randomly generated overlay identifiers. As long as the identifiers of nodes and objects are uniformly distributed in the namespace, the traffic due to storage and retrieval of objects is also uniformly distributed on nodes in the network. However, in the context of MP2P systems, randomly, distributed node identifiers introduce several drawbacks. On one hand, the virtual neighborhood of a node would be uniformly distributed in the deployment area. As a result, update messages that are required to provide fresh routes to the virtual neighbors have to be sent over multiple underlay hops in medium and large scale scenarios. On the other hand, the traffic generated by a lookup is affected whenever node identifiers are generated randomly in an MP2P system. In order to complete a lookup successfully, intermediate nodes are required to forward the request message to the destination. Due to randomly generated node identifiers, those intermediate nodes are uniformly distributed in the deployment area. Therefore, a request message may have to be sent across the deployment area even when the destination is geographically close to the source of the lookup as shown in Figure 6 on the previous page.

During a lookup, a request message is generated and forwarded according to the DHT's routing algorithm. As a result, the message is getting logically closer to its destination with each hop. By harnessing the geographical position of the

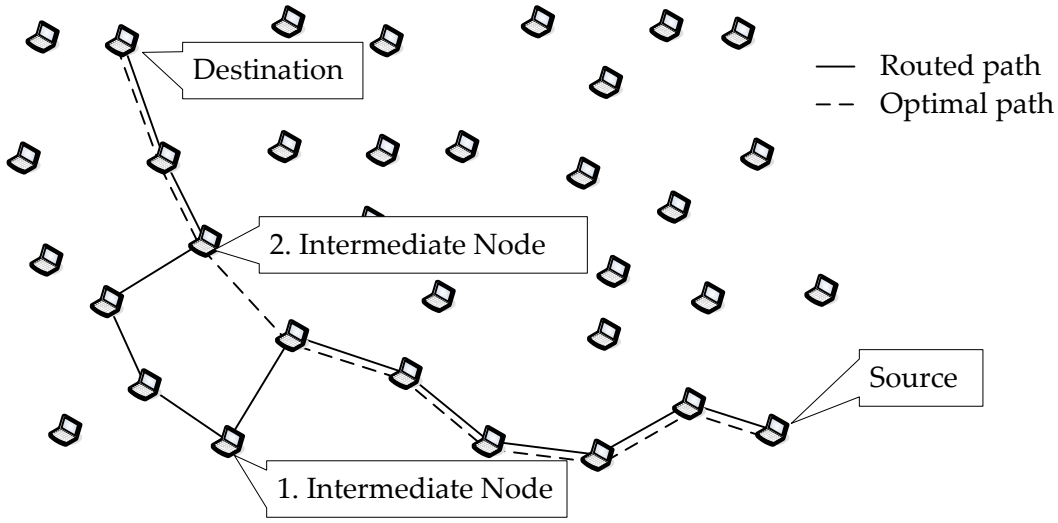


Figure 7: Example for routing of a location aware Mobile Peer-to-Peer system

nodes during the generation of their identifiers, their geographical location can be mapped on their virtual location in the namespace of the overlay identifier. As a result, with each hop a lookup message is forwarded to a node that is logically but also geographically closer to the destination. Therefore, the average number of underlay hops per lookup can be strongly reduced. An example for this geographically adapted lookup is shown in Figure 7. Furthermore, as all virtual neighbors are as well geographical neighbors, the traffic generated due to the maintaining links to those virtual neighbors (as required by several DHTs) is strongly reduced. However, as mobile nodes were assumed, the node identifiers have to be adapted whenever a node is moving in the deployment area. In order to adapt this approach to a mobile scenario, it seems to be more promising to harness a geographical area instead of the specific location of a node to define the overlay identifier. As long as the node is within this area, the node identifier does not have to be adapted. Yet, as multiple nodes may join the same geographical area, the node identifier must not be only a function of the location but has to be a combination of a randomly generated and a location based identifier. Each of those geographical areas are attached to a specific virtual area within the identifier-namespace of the DHT. Those areas, that maps the virtual structure of the overlay to the geographical area are henceforth called clusters. As a result, nodes within such a cluster are both virtual as well as geographical neighbors. Furthermore, the arrangement of the clusters is also crucial as adjacent clusters have to be mapped to adjacent areas in the virtual namespace.

Due to the dynamic topology, it can be assumed that the routing tables of the MP2P system become obsolete frequently. On demand updates as proposed by several DHTs is, therefore, not feasible in the context of an MP2P scenario.

Thus, periodical routing tables updates are required to ensure the freshness of the stored entries even in a highly dynamic scenario. Otherwise, a high fraction of outdated routing table entires could affect the efficiency of the MP2P system. Furthermore, to ensure the availability of objects even when single nodes leave the network or are temporary unavailable due to the dynamic topology of the underlay, replicas are required.

3.3 UNDERLYING SYSTEMS AND ARCHITECTURE

Our Clustered Pastry system inherits the major characteristics of the underlying protocols and algorithms. In particular, the MANET underlay and the P2P overlay protocol have a strong influence on the resulting system, even though those underlying protocols have to be adapted according to the requirements and challenges of the decentralized mobile environment. Furthermore, the way of how to integrate those protocols into the MP2P system has a major influence on the characteristics and the efficiency of the resulting Clustered Pastry system.

In this section, merit and flaws of available underlying protocols are discussed. As a result of this section, we define essential design components of our Clustered Pastry system.

3.3.1 *Mobile Ad hoc Underlay*

The underlay enables the communication between the peers in the network. Therefore, a MANET routing protocol is required to establish routes between participating nodes. As stated before, a MANET underlay provides less resources in terms of bandwidth compared to the traditional underlay of a P2P network (the Internet). However, the available bandwidth as well as the reliability of the provided underlay services depend partially on the underlay protocol (as shown by Seither et al. [95] in an exemplary way). Due to this fact, the underlay routing protocol has a high impact on the efficiency of the resulting Clustered Pastry system.

As discussed in Section 2.1.2 many MANET routing protocols have been developed in the last decade. Most of those architectures have been evaluated theoretically or by simulation only. As shown by [70] and [38] even the efficiency and reliability of well-known protocols as AODV is degraded when used in realistic testbed environments due to effects, that had been neglected by the simulator based evaluations. OLSR is one of the few routing protocols that has been evaluated in real world scenarios. This protocol has been used as communication substrate in a disaster management exercise as part of the DUMBO [57] project and by the Freifunk network [1] to provide Internet connectivity to households that are disconnected from the wired Internet. Especially when used in combination with a link quality based metric as ETX [23], fairly reliable transmissions can be achieved compared to hop based protocols as AODV. Yet, lost or dropped messages have still to be considered due to the characteristics of the wireless channel. Furthermore, due to the periodically sent *hello* messages, each node that uses OLSR is aware of nodes that are within transmission range.

Therefore, OLSR ETX is used as underlay routing protocol for this novel Clustered Pastry system. Yet, this system can be adapted with little effort to use other underlay routing protocol including reactive protocols, e.g., AODV or DSR [56].

3.3.2 *Peer-to-Peer Overlay*

The overlay provides services as storing, retrieving, and maintaining objects in a decentralized way. Due to the MANET underlay and the clustered structure of the resulting Clustered Pastry system, the following three requirements arise for the MP2P overlay. (I) The routing algorithm of the P2P system has to be adaptable to the

location aware structure of the network. Otherwise, the MP2P system cannot benefit from the geographical location of the participating nodes. Furthermore, (II) the traffic generated by the overlay has to be minimized due to the bandwidth limitations of the MANET underlay. This includes traffic introduced by the lookup mechanisms as well as traffic due to routing table maintenance. Finally, we assume that each node provides the same resources. Therefore, (III) no centralized or semi-centralized entities are available in the MP2P network. In the following paragraphs, we determine on which P2P system the overlay of our MP2P is based on.

As mentioned in Section 2.1.3, two basic P2P architectures, structured and unstructured, have been proposed that enable these services. Unstructured P2P systems can further be classified according to their architecture in centralized, pure and hybrid P2P systems. Centralized systems do not fit the requirements of our scenario as they require a central entity, that has to be available in order to use the services provided by the network. However, pure P2P systems do not require a predefined infrastructure and, therefore, can also be deployed in mobile, decentralized scenarios, e.g., disaster relief. As a result, they seem to be more promising when considering an MP2P system. Yet, these approaches harness a flooding based routing mechanism. This results in a high signaling overhead whenever a lookup is initiated [103]. Due to the limited bandwidth introduced by the underlay, traffic overhead has to be minimized. Therefore, pure P2P based overlays are not considered for developing an MP2P system in this thesis. Considering hybrid P2P architectures, lookup request messages have to be sent solely to the superpeers. As a result, no flooding is required and, therefore, overhead induced by the lookup mechanism can be strongly reduced [5]. Thus, hybrid systems satisfy most of the previously discussed requirements for MP2P overlays. Yet, a structured system is preferred in the context of this thesis, as DHTs neither have to rely on a costly flooding based routing mechanism, nor require highly available semi-centralized entities as superpeers [103].

Most DHTs provide a similar functionality and are comparable in their basic design as well. At the first glance, a CAN based overlay seems to be promising for a geographically structured architecture. As mentioned in Chapter 2.1.3 CAN is based on a hypercube structure. By using a two dimensional namespace CAN could be harnessed to map the geographical area directly on the virtual namespace. However, this approach introduces several drawbacks. As nodes were not only mapped directly to a point in the virtual space but also to an area, leaving or joining nodes would also affect the node identifiers of adjacent nodes. Therefore, the required updates of the overlay identifiers would result in additional traffic overhead. Furthermore, the efficiency of a CAN system depends directly on the dimensions of the hypercube structured namespace. A two dimensional CAN provides, therefore, only a strongly limited efficiency.

The hybrid Pastry DHT on the other hand does not provide a direct mapping of coordinates to overlay identifiers. Yet, this architecture can be adapted by using geographical coordinations of a node during ID generation. In addition, Pastry satisfies the three requirements discussed earlier and introduces multiple benefits, that can be harnessed in order to develop an efficient location aware MP2P architecture.

- Pastry uses a prefix based routing algorithm. With each overlay hop, another prefix digit is matched and, therefore, the virtual distance to destination is

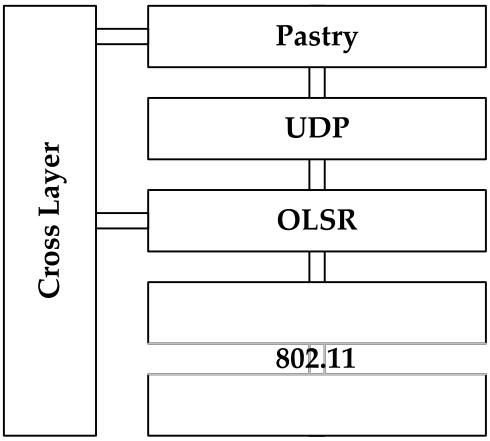


Figure 8: Layer model of the Clustered Pastry approach

reduced. This can be exploited to provide a location aware MP2P routing mechanism as will be discussed in the following section.

- The parameter b specifies the size of Pastry's routing tables. Therefore, the number of *routing table* entries can be easily adapted to a specific scenario. Due to this fact, the structure of the *routing table* does not have to be modified in order to influence the number of *routing table* entries and the resulting traffic due to update messages.
- Due to the structure of Pastry, no centralized or semi-centralized nodes are required.
- Pastry maintains a *leaf set* that stores links to virtual neighbors. Those virtual neighbors are crucial in order to provide reliable lookup services. In the context of a location aware architecture, virtual neighbors are also geographical neighbors. As a result, this routing table can be efficiently maintained with low effort.
- Multiple MP2P systems have been developed recently that are based on Pastry. As security mechanisms for the resulting MP2P system are developed in the second part of this thesis, the probability is increased that those mechanisms can be adapted to other MP2P systems that are based on Pastry.

As a result, we use Pastry as substrate to develop the overlay of our Clustered Pastry system. However, the clustered design concept can also be used in the combination with other overlay architectures.

3.3.3 *Mobile Peer-to-Peer Architecture*

As mentioned in Chapter 2.1.4, three basic groups of MP2P systems exists (*integrated*, *layered*, and *cross-layered* MP2P systems). *Layered* MP2P systems are limited in their degree of adaptation to the challenges that arise when combining a MANET underlay with a DHT overlay. The overlay is not able to benefit from the information that could be provided by the underlay due to layering. Only minor modifications in the overlay

may be applied in order to reduce the traffic overhead and to ensure the availability of the provided services.

On the other hand, *integrated* MP2P systems combine the overlay and the underlay protocols at a single layer. As a result, those MP2P systems are able to harness location based or overheard information to improve their routing and update functionality. Therefore, *integrated* MP2P systems are very efficient. Yet, due to the combination of underlay and overlay at a single layer, the underlying protocols have to be strongly adapted. Due to this fact, the algorithms of *integrated* MP2P systems can vary widely regarding the implementation of the routing and lookup algorithm. Thus, mechanisms that are developed to increase the robustness of a specific *integrated* system can (mostly) not be deployed to increase the robustness of any other MP2P systems.

Therefore, this novel MP2P system is based on a *cross-layered* architecture. Due to cross layering, the overlay benefits from information provided by the underlay. Yet, overlay and underlay protocols are implemented at separate layers. As a result, underlay and/or overlay protocols may be replaced by other protocols in order to match the requirements of a specific scenario with little effort. Furthermore, security mechanisms developed for a cross layered MP2P system can mostly be used in combination with other MP2P systems.

In summary, a *cross-layered* architecture enables the overlay to exploit information provided by the underlay. Furthermore, both of the underlying protocols can be replaced when required by the scenario. Therefore, they combine the benefits of the *layered* and *integrated* architectures. Due to this fact, we will develop our Clustered Pastry system based on a cross-layered architecture.

3.3.4 *Layer Model of the Clustered Mobile Peer-to-Peer Approach*

After discussing the basic overlay and underlay protocols, that are used as underlying architecture for our Clustered Pastry system, a basic layer model can be built. As shown in Figure 8 the underlay is built on top of an 802.11 standard [3] that defines both, the geographical as well as the data link layer. 802.11 is a set of standards that is widely used for wireless communication and includes, e.g. 802.11g or 802.11n. However, MP2P systems can be used as well in combination with other underlying wireless technologies, e.g., Bluetoooth [14]. Furthermore, the User Datagram Protocol (UDP) [82] is used as transport layer protocol. UDP is a connectionless, message-oriented protocol that does neither provide congestion control mechanisms nor manages packet retransmissions. Therefore, this has to be handled at the application layer by the MP2P system. Due to the layering concept discussed in previous section, cross layering between the underlay (network layer) and the overlay (application layer) is required.

3.4 STRUCTURE OF THE CLUSTERED PASTRY SYSTEM

Overlay identifiers are used to determine the virtual distance between the nodes in a structured P2P network. This virtual distance is harnessed during the overlay routing to select the next hop node. As a result, the overlay identifier assignment directly affects the route of a request message. Therefore, the adapted identifier allocation method as well as the influence of mobility on these overlay identifiers are discussed

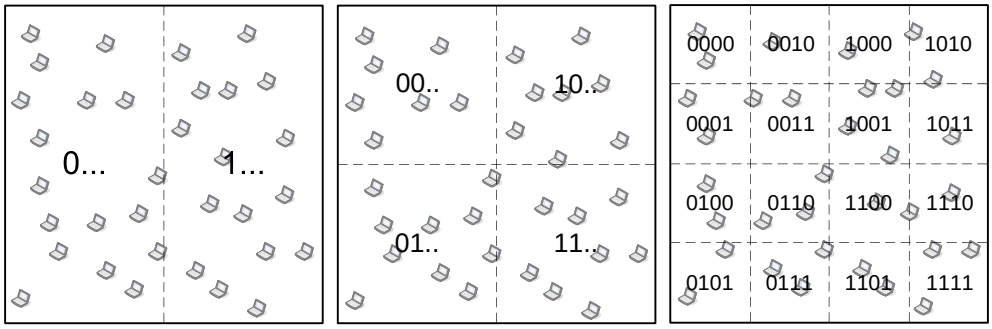


Figure 9: Clustering the deployment area in 2, 4 and 16 clusters

in the following section extensively. Furthermore, the structure of the overlay identifier is introduced and discussed.

3.4.1 *Clustering*

As mobile nodes have to be considered in a disaster relief scenario, mapping the overlay identifiers to the exact geographical position of a node introduces a major challenge. Whenever a node is leaving its position due to mobility, the overlay identifier has to be adapted. The resulting updates of routing tables and redistribution of stored objects introduce a high traffic overhead. Therefore, it is more beneficial to harness the rough area a node is located in for generating the overlay identifier. This results in a static overlay identifier as long as the node is located in the same geographical area, being called henceforth a *cluster*.

Pastry and, therefore, the novel MP2P system uses a prefix-based routing algorithm. Therefore, the first significant digits of an overlay identifier do not only affect the virtual position of a node in the namespace, but are also crucial for the systems routing mechanism. Due to this fact, the prefixes of the overlay identifiers are used to map a node to a cluster. Each node located in the same cluster has to provide an overlay identifier with an equal prefix (defined by the cluster) and a randomly generated suffix.

Considering a disaster relief scenario, it can be assumed that the size of the deployment area is known to the technical operational command. Therefore, this area can be split in multiple clusters as shown in Figure 9. These clusters may either be generated automatically based on the deployment area or defined by the operational command group according to the characteristics of the specific scenario. Furthermore, the arrangement of these clusters is essential. Neighbor clusters have to provide a similar prefix. For this reason, it can be assured that the location of the destination is narrowed down with each hop. For example, consider a node with the prefix 1010 that initiates a routing for a node with the prefix 0000. The first overlay hop would at least provide a node that matches the first digit. As shown in Figure 10, the whole right half-plane of the deployment area can be neglected for the rest of this lookup after this first hop. After the second hop, at least the first two digits have to match the destination's overlay identifier. Hence, the destination node has been narrowed down to an area that is less or equal to a quarter of the deployment area. For this reason, we assume that the location aware routing mechanism is able to provide efficient services in terms hop distance and, therefore, traffic.

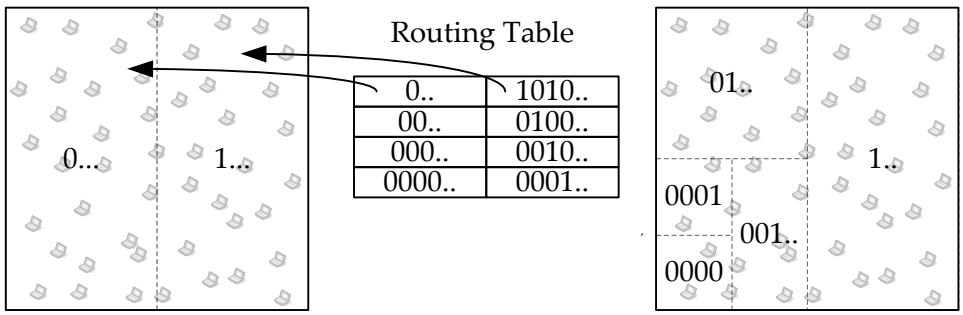


Figure 10: Structure of the clusters and the *routing table*

Obviously, the structure of the clusters and the overall number of clusters affects the MP2P systems routing mechanism and, therefore, the traffic introduced by a lookup. Both of those characteristics are defined by two parameters, the number of clusters per level (p) and the number of cluster levels (l). The clusters per level are defined by the maximum value of the overlay identifiers digits. An identifier that is based on binary digits would, e.g., result in two clusters per level. Hence, an identifier to the base of four would result in four clusters per level, a hexadecimal identifier in sixteen clusters per level and so on. As discussed in Section 2.1.3, the digit size and, therefore, the number of clusters per level, is defined by the parameter b. On the other hand, the number of cluster levels depends on the prefix size (s_{prefix_size}) of the overlay identifier. For each digit of the prefix, another level of clusters is introduced. Due to this fact, the overall number of clusters ($n_{clusters}$) is defined by the multiplication of the number of clusters per level with the cluster level as shown in Equation 3.1. Both, the parameter b as well as the prefix size further define the structure of the *routing table*. Hence, the structure of this table correlates to the structure of the clusters.

$$n_{clusters} = p * l = 2^b * s_{prefix\_size} \tag{3.1}$$

Considering the example shown in Figure 10, a scenario is assumed that is based on a binary overlay identifier ($b = 1$) and a prefix size of four digits. In the middle of this figure, the *routing table* of the node with the ID prefix 0000 is shown. On the left side, the resulting number of clusters per level is illustrated. On the right side of this figure, the resulting levels per cluster are shown. Due to the binary identifier digits, each routing table row stores two entries only. As p is equal to the number of entries in a table row, the basic cluster architecture is based on two clusters). According to the 4 table rows, 4 levels of clusters are required. However, due to the structure of the *routing table*, it is not required to provide links to every cluster (as will be discussed in Section 3.5).

3.4.2 *Overlay Identifier*

Overlay identifiers are used by DHTs to identify nodes and objects that have been uploaded to the network. In the following paragraphs, the overlay identifiers as harnessed by Clustered Pastry are discussed in detail.

OVERLAY NODE IDENTIFIER

Each node in the network is identified by a unique overlay identifier. These node identifiers are composed of the cluster-based prefix and a randomly generated suffix. As extensively discussed in the previous paragraph, the prefix is used to harness the geographic location of a node to optimize the DHTs routing algorithm. The suffix of the identifier is required to be able to address a specific node within a cluster. In a worst case scenario, every node is located within the same cluster. Even in such a scenario, it is essential that node identifiers are unique. Therefore, the size of the suffix must be sufficient large in order to ensure a collision free distribution of identifiers. Pastry uses overlay identifiers with a size of 128 bits by default. Thus, up to 2^{128} addresses are available and the probability that the identifiers of two nodes are equal is quite low (e.g., less than 0.0001 % in a scenario with 1.000.000 peers). However, those identifiers are included in every lookup request or update message. Therefore, the identifier size has to be scaled down in order to reduce overhead generated by those messages. Yet, the size of the overlay identifier has still to be large enough to ensure a collision free distribution of overlay identifiers. Due to this fact, we defined an overlay identifier with the overall size (including prefix and suffix) of 32 bits. As we assume a decreased number of participants in our MP2P scenarios, the probability of a collision is still below 0.005 % considering a scenario with 200 nodes, a suffix size of 30 bits, and a prefix size of 2 bits.

Whenever a node leaves a cluster, the node identifier has to be adapted. Therefore, at least the prefix of the node ID has to match the new cluster's prefix. However, the suffix itself may not be modified, but can be constant. Furthermore, the routing tables of the node that has changed its identifier as well as the routing tables of those nodes that have stored a link to this node have to be updated.

Clustered Pastry may further harness context awareness, when required. Therefore, an extended identifier may be used in order to define the nodes affiliation. This identifier adds further digits between the prefix and the suffix. Those digits can be used to identify a peer as a member of, e.g., the firefighter or the ambulance. This extended identifier can be used in combination with the prefix to send an anycast request to a peer that is within a specific cluster and provides specific services. An example scenario could include the detection of a nearby (within the same cluster) ambulance when medical services are required. However, due to the affiliation based extension of the node identifier, the overall size of this identifier may have to be increased in order to prevent collisions.

OVERLAY OBJECT IDENTIFIER

Beside nodes, also objects are identified by unique 32 bit overlay identifiers. As proposed by the Pastry DHT, object identifiers in Clustered Pastry are generated by hashing the objects name. Each objects is, thereafter, stored at a node that is virtually closest to the object identifier. However, in the Clustered Pastry system, an object is always stored within the cluster that provides the same prefix than the object's identifier. Within this cluster, the node that is logically closest to the overlay identifier has to maintain this object.

For example, we assume that an object with the identifier 10111 has to be stored in the MP2P network and two nodes are available in the virtual neighborhood of the objects identifier. Node A has the same prefix and has the identifier 10100. Node B with the identifier 11000 is logically closer to the object but differs in the prefix. Due

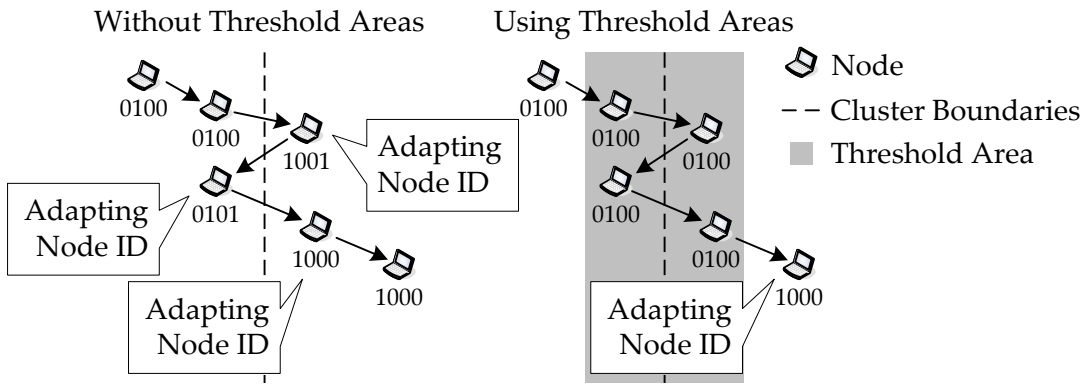


Figure 11: Node mobility and the resulting adaptation of identifiers with and without threshold areas

to Pastry's routing mechanism, the object would be stored at node B as this node is the logically closest node to the object. Clustered Pastry stores this object at node A as this node shares the same prefix and is the node that is logically closest to the object in this cluster.

However, in order to harness location awareness, object identifiers may also use the prefix of the cluster, they are attached to. For this reason, e.g., sensor information about a specific area may be directly stored within this area as most probable this information will be requested from a node within this cluster. Yet, we consider the location awareness regarding the object's position as optional. Therefore, this functionality will not be evaluated within this thesis.

THRESHOLD AREA

Nodes have to adapt their overlay identifier whenever they leave and/or join a cluster. Even though, those nodes do not leave nor join the network but are only changing their cluster, the resulting effects on the network are similar to those resulting from churn. Routing tables have to be updated that provide links to the changing node as well as the routing table of this node itself. Furthermore, objects that are stored at the node that is leaving a cluster have to be redistributed. Thus, traffic overhead is generated whenever a node changes its cluster and, therefore, has to adapt the node identifier. In order to minimize the resulting traffic overhead, the number of cluster changes has to be minimized.

Due to mobility, nodes may alternate their cluster and, therefore, their overlay identifier frequently when they move along the border of two clusters. This results in an unnecessary traffic overhead as each node identifier may be valid for a very short amount of time only. An example is shown in Figure 11 on the left side. A node with the identifier 0100 moves toward and crosses the border. As a result, the node identifier has to be adapted. While moving along the border, this node changes the cluster two more times within a short amount of time, each resulting in overhead due to adapting the node's identifier, updating the routing tables, and the redistribution of stored objects. In order to avoid such an unnecessary traffic overhead, threshold areas are introduced. Whenever a node crosses the border between two clusters, this node does not adapt its overlay identifier directly, but only when this node has also crossed the border's threshold area. As shown in the example on the right side of Figure 11, this threshold area is arranged along the clusters border. Even though the

node in this example crosses three times the border in a row, the identifier is only adapted when this node leaves the threshold area. For this reason, cluster related churn and the resulting traffic overhead can be reduced.

The size of these threshold areas depends on several parameters. On one hand, the size of the clusters and the average node speed has to be considered when defining the threshold area. When assuming highly dynamic scenarios, e.g., in vehicular networks, large scale threshold areas are required. However, when these areas are too large, nodes may lose the connection to their virtual neighbors. This may result in stale *leaf sets* as these are updated via a direct connection to the virtual neighbors (as discussed in the next subsection). Therefore, the upper bound of the threshold's size is defined by the transmission range of the networks participants. The specific size of the threshold areas is defined and evaluated in Chapter 4.3.

3.5 ROUTING IN THE CLUSTERED PASTRY SYSTEM

The routing mechanism of an MP2P system is essential for the services provided by the network. Furthermore, this mechanism has a strong impact on the traffic introduced by the system including traffic due to maintenance and lookups. In this section, the functionality of Clustered Pastry's routing mechanism is introduced and discussed. Furthermore, the required routing tables are discussed in detail and update mechanisms are introduced.

3.5.1 *Routing Tables*

Clustered Pastry is based on a Pastry overlay and an OLSR protocol in the underlay. Both of these underlying architectures require routing tables in order to provide routing functionality.

The MANET protocol OLSR uses on a single routing table. This table provides underlay routes to the nodes in the network. Pastry on the other hand is based on three different routing tables (as discussed in Chapter 2.1.3). Those tables are used to map overlay identifiers to underlay addresses. However, to meet the challenges introduced by a mobile, decentralized scenario and due the clustered system, most of these routing tables have to be adapted. As OLSR is not affected by the clustered structure of the network at all, only Pastry's routing tables have to be modified. In the rest of this section, the required modifications on the overlay routing tables are discussed.

ROUTING TABLE

Clustered Pastry's *routing table* provides links to a set of nodes that are located within different clusters. This table is prefix-based and tree structured, similar to the *routing table* of Pastry (see also Chapter 2.1.3), and is structured as follows:

- Each row n of Clustered Pastry's *routing table* provides links to nodes that match on $(n-1)$th prefixes but differ on the nth prefix.
- The number of rows in the *routing table* is equal to the size of the overlay identifier's prefix.

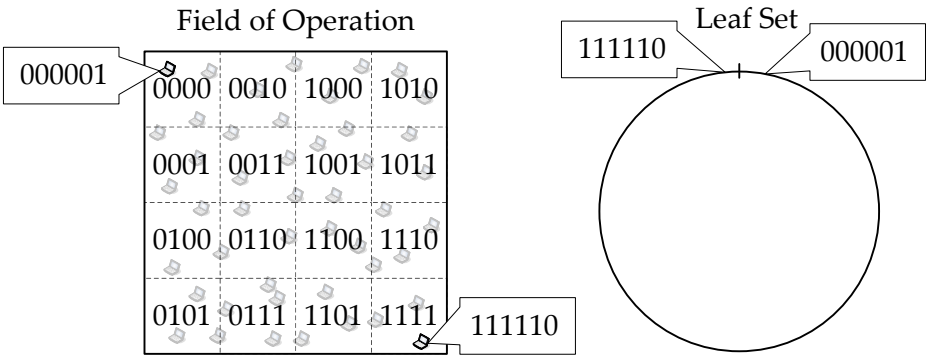


Figure 12: Example for a virtual neighbor in a clustered environment

- Each column provides a link to nodes that shares the same $(n-1)$th prefixes but are located in different clusters. One entry of each column always refers to the owner of the *routing table*.
- The number of required columns per row is defined by the clusters per level.

As a result of those characteristics, the structure and size of the *routing table* is closely related to the structure of the clusters.

An example for the structure of a *routing table* of a node located in the cluster with the prefix 0000 is shown in Figure 10. Due to the binary structure of the node identifier, two entries per row are required. As a result of the prefix size of four digits, four rows are stored within the *routing table* in this example.

LEAF SET

The *leaf set* provides links to nodes that are located in the virtual neighborhood of the owner of this routing table. As the overlay identifiers are generated as a function of the geographical position of the nodes, each node in the *leaf set* is also a geographical neighbor. However, the ring topology of Pastry's *leaf set* may not be beneficial to the Clustered Pastry's *leaf set*. Whenever the virtual neighbor is located within another cluster, overhead is introduced as links to distant nodes have to be maintained. An example for a worst case scenario is shown in Figure 12. The virtual neighbor of the node 0000010 that is located in the upper left cluster is located on the lower right end of the network. Due to this fact, routing to a virtual neighbor would result in sending a message through the whole deployment area. Therefore, the structure of the *leaf set* is modified to ensure that virtual neighbors are always located within the same cluster. Furthermore, each cluster forms an own *leaf set* ring. This ring includes only nodes that are located in the same cluster. In contrast to Pastry, the *leaf set* size is not defined by a fixed parameter, but by the number of nodes in the cluster. As a result, each node is aware of every other node that is located in the same cluster.

NEIGHBORHOOD SET

Beside the *leaf set* and the *routing table*, Pastry also provides a table including nodes that are geographically close. However, this *neighborhood set* set is not used during routing. Yet, this set can be used after a node disconnects from the network to provide a list of bootstrapping nodes. In the context of the Clustered Pastry system, no *neighborhood set* is required, as the *leaf set* does not only provides links to nodes that

parameter b	1	2	4
columns	2	4	16
Routing table rows	4	2	1
Routing table entries	4	6	15
Average hops	3.00	2.50	1.94

Table 1: The influence of parameter b on the *routing table* and the routing in a scenario with 16 clusters

are logically but also geographically close. Therefore, functionality provided by the *neighborhood set* can also be provided by the *leaf set* in a Clustered Pastry system. As the *leaf set* is required by the routing mechanism and is able to provide the *neighborhood set's* functionality without introducing further overhead concerning routing table updates, no *neighborhood set* is used in the Clustered Pastry system.

NUMBER OF CLUSTERS PER LEVEL

The number of Clusters per level, defined by the parameter b, strongly affects the structure of the Clustered Pastry system, as previously discussed. Both, the structure of the *routing tables* as well as the number and the arrangement of the clusters in the network is defined by this parameter. Therefore, the value for the parameter b has to be chosen wisely.

Pastry's default value of the parameter b is four. This results in identifier digits that are based on hexadecimal values. Furthermore, each *routing table* row provides 16 entries and when considering a Clustered Pastry system, the basic cluster rate would be 16 clusters per level. However, this also results in 15 addresses per row that have to be maintained frequently. Yet, considering a scenario with a low value of b, the number of *routing table* entries per row and, therefore, the number of required updates is reduced. However, as also the number of clusters per level is decreased, the number of levels has to be increased in order to provide the same amount of overall clusters. This result in an increased average number of overlay hops that are required for a lookup.

An example for a setting with 16 clusters is shown in Table 1. As previously assumed, the average number of hops is increased whenever the value of the parameter b has been reduced. However, the number of overall entries, that are required to provide a *routing table* in this setting is strongly reduced when using a low value of b. As those entries have to be updated frequently, the overall traffic generated by a low value of b is decreased compared to a scenario with a high value of b. Therefore, and when considering that routing table entries have to be updated repeatedly per minute while lookups are only initiated every few minutes, the advantages of a low value of the parameter b outweigh the disadvantages.

In order to minimize the traffic due to routing table updates, Clustered Pastry's default value of the parameter b is set to two. As a result of this setting, the overlay identifier is based on binary digits. Furthermore, the *routing tables* provide two columns per row and two clusters are provided per level.

3.5.2 *Updating the Routing Tables*

Whenever a node leaves or joins the MP2P network or a cluster, the routing tables of this node as well as each routing table that provides a link to this node has to be updated. However, update messages are sent by Pastry only when either a node has joined the network or as a result of a failed lookup. Updating routing tables as a part of the bootstrapping mechanism is required in order to provide a valid *routing table* to the booting node and to update the *leaf set* of the virtual neighbors of this node. Assuming a broken link whenever no reply to a request message is received is reasonable in a wired network. As messages will be delivered with a very high probability to the destination node when sent via a wired network, a lost message mostly indicates that an intermediate node has left the network and, therefore, cannot anymore be addressed. However, in wireless scenarios, messages may get lost due to the characteristics of the wireless channel (e.g., as a result of collisions). Furthermore, an on demand update mechanism as proposed by Pastry is not sufficient, when considering an MP2P network with a highly dynamic topology. Otherwise, idle nodes neither sent nor forwarded a message for an extended time interval would not be able to detect broken links. This would result in stale *routing tables* and, therefore, affect the availability of the networks storage and retrieval services. For this reason, periodic updates are required in order to ensure fresh routes for Clustered Pastry's *routing tables*.

The *leaf set* is harnessed to determine the precise destination of a request message during a lookup. Therefore, it is essential that the routing entries that are provided by this table are valid. This can be assured by maintaining periodic updates. Due to the clustered structure of the MP2P system, *leaf set* nodes are located within the geographical neighborhood of a node. Based on this characteristic, underlay messages are exploited to update the *leaf set* in a Clustered Pastry system. *Hello* messages are sent periodically by the OLSR protocol in order to detect nodes that are within transmission range (as mentioned in Chapter 2.1.2). Those messages can be extended by appending *leaf set* information including the IP address, the overlay identifier, and the lifetime of this dataset. Hence, update messages are broadcast every 2 seconds by every node in the cluster. As a result, a fresh and reliable *leaf set* is ensured. Yet, this update mechanism introduces an increased traffic overhead. This will further be discussed in Chapter 3.5.3.

In order to provide valid *routing table* entries, each stored link has to be validated via a periodically sent update message. Whenever a node receives such an update message, the requested overlay identifier is compared to the own identifier. If these identifiers match, a reply message is sent in order to validate this *routing table* entry. Otherwise, the receiver's *routing table* is used to provide a valid node for the requested table entry. If available, the proposed entry is attached to the message that is replied to the source node of this update request. However, if either no reply message is received or the reply message does not contain a valid entry, a virtual neighbor of the source node is harnessed to update this *routing table* entry.

Furthermore, the average lifetime of a *routing table* entry can be extended by harnessing the threshold areas. Whenever a node that is located within a threshold area receives an update request, this node does not reply its own routing information but the overlay identifier and IP address of another node in this cluster. This information

can be easily retrieved by using the *leaf set*. Due to this mechanism, routing entries that refer to nodes that may change their cluster soon are replaced automatically.

Another challenge is introduced by nodes that join the network at a cluster that is still empty. Those nodes create their own routing tables by contacting a node that is located in an adjacent cluster. As part of this bootstrapping procedure, at least the routing tables of the bootstrapping node as well as the tables of the logically closest neighbor are updated. However, all other nodes in the network are unaware of this node. Yet, at least the nodes that are located in the logically closest cluster require information about the booted node in order to provide routing services. Therefore, information of newly detected nodes is also broadcast in the cluster via the *hello* messages for a limited amount of time.

3.5.3 *Reducing the Overhead Introduced by the Update Mechanism*

Harnessing the OLSR *hello* messages to update the *leaf set* provides periodically fresh links to virtual neighbors. Even though this is essential for a reliable lookup mechanism, a high amount of traffic overhead is introduced due to this update mechanism as each virtual neighbor within this cluster increases the size of the *hello* message by 9 bytes. Therefore, we propose multiple mechanisms to reduce the traffic overhead that has been introduced by the update mechanism in the following paragraphs. These approaches are either based on adapting the *update frequency* or reducing the *data amount* attached to each *hello* message.

ADAPTING THE UPDATE FREQUENCY OF THE LEAF SET

A straightforward approach to reduce the traffic overhead generated due to the *leaf set* updates is based on reducing the *update frequency* of this routing table. However, this update frequency depends on the OLSR neighbor detection mechanism and, therefore, cannot be arbitrarily set. As a result, the *leaf set* update frequency can only be raised to a multiple of the frequency of the *hello* messages. However, in order to reduce the traffic overhead but also avoid a stale *leaf set*, the update frequency has to be selected carefully.

REDUCING THE SIZE OF THE UPDATE MESSAGES

Besides reducing the update frequency, the *data amount* that is attached to the *hello* message can also be minimized to reduce the traffic overhead. Either the size of the data set (including IP address, the overlay identifier and the TTL) per leaf or the number of attached sets can be reduced.

Each transmitted data set consists of the overlay identifier, the IP address, and the TTL of a virtual neighbor. Those data sets bear redundancy, as the prefix of the overlay identifier or the network identifier of the IP address. In order to update the *leaf set*, each overlay identifier that is attached to the *hello* message has to share the same prefix. As a result, it is sufficient to transmit the identifiers suffix only. On the other hand, only a limited amount of nodes have to be assumed in an MP2P scenario. Therefore, a class B IP network (up to 65,534 nodes) is more than sufficient to address all peers in the network. Due to this fact, neither the leading bits nor the identifier of the network are required to provide update functionality as every node is located within the same network. Considering these optimizations, the size of each update

data set can be reduced by 33 % to 6 bytes when assuming a prefix size of 1 byte. Furthermore, the overall size of the data sets can be reduced by harnessing data compression mechanisms. Yet, due to compression, single bit errors during the data transmission over the lossy wireless channel would not affect a single update set but in the worst case all sets that have been attached to this *hello* message. Therefore, no compression mechainsims are considered in the context of this thesis.

On the other hand, the number of data sets that are attached to *hello* messages can be reduced. This can either be done by considering only nodes that have recently joined or left the cluster or by using gossiping mechanisms. Considering only nodes for the *leaf set* update that have recently joined or left this cluster seems to be a promising approach to reduce the overhead in the network. As only few nodes change their cluster on average within the update interval, only few data sets have to be broadcast. However, this approach has two major drawbacks. Nodes that join the network are no longer able to build their *leaf set* based on received update messages but have to contact a node in the cluster. On the other hand, messages may get lost in a wireless network. Hence, a lost notification would result in outdated *leaf sets*. Even worse, those stale *leaf sets* entries can only be updated with high effort. As a result, it is not recommended to update the *leaf set* by only considering nodes that have left or joined the cluster recently.

However, gossiping can be used to reduce the number of data sets that are transmitted with each *hello* message. Gossiping is a mechanism often used in, e.g., MANETs to reduce traffic introduced by the routing mechanism [120] [42] [28]. Whenever a node receives a message that has to be forwarded via broadcasting, this message is only transmitted with the probability p. As in most cases multiple nodes have to broadcast this message that are within the same transmission range, the message is still distributed efficiently. A similar mechanism can be harnessed in the context of the MP2P system in order to reduce overhead generated by the update mechanism. Whenever a *hello* message has to be sent, each data set is only attached to this message with the probability p. Also a maximum number of n_{DS} data sets per *hello* message can be defined in order to minimize the traffic overhead. This approach is able to adapt the traffic generated due to the update mechanism to the node density. In a low density scenario when less than n_{DS} nodes are located in a cluster, the whole *leaf set* is attached to the *hello* message. Yet, in a scenario with high node density, only a randomly selected fraction of the nodes in the *leaf set* is attached to the *hello* message and hence the traffic generated by the update mechanism is limited. This makes particular sense as overall traffic is increased in high density networks anyway. Furthermore, in most cases low density networks provide sufficient bandwidth but may suffer due to gossiping as the probability for stale routing tables increases when only few nodes are within direct transmission range.

PROXIMITY METRIC

As mentioned in Chapter 2, Pastry harness a proximity metric when updating the *routing table*. Due to the structure of this table, multiple nodes suffice the requirements of each table's entry. Therefore, this proximity metric is used to detect those nodes that are geographically closest to the owner of the table. These nodes are thereafter stored in the *routing table*.

This seems to be a promising approach also in the context of MP2P systems. The proximity awareness of the *routing tables* would result in preferring links to nodes

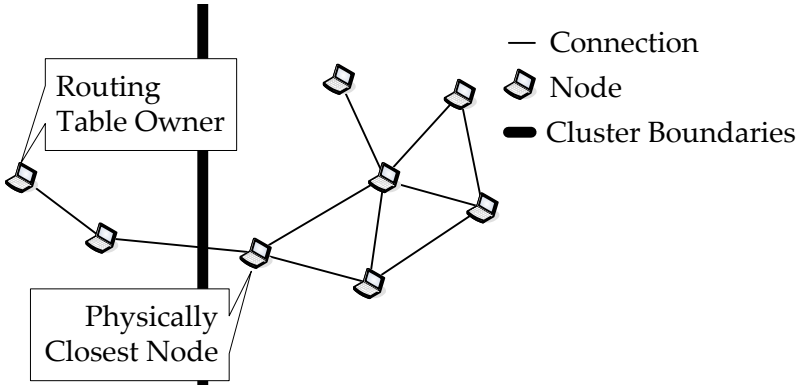


Figure 13: Preferred nodes in a scenario with proximity metric

that are geographically close. Therefore, the average number of underlay hops per lookup could be reduced. Furthermore, the proximity of a node could be determined more reliable by using cross layer information from the underlay.

However, the proximity metric introduces three major drawbacks to the Clustered Pastry system. The proximity metric can be harnessed to (I) increase the impact of several attacks as will be discussed in detail on Chapter 5.3.4. Furthermore, (II) traffic overhead is introduced due to the proximity detection mechanism. And (III) the average lifetime of an entry is reduced due to this metric.

Pastry's proximity metric determines the distance to a node by harnessing the round trip time of ICMP messages. Due to this fact, a ping message has to be sent to each node that is considered as a *routing table* entry. This would result in high overhead in a highly dynamic network as each node in the network would have to contact a large set of nodes frequently.

Furthermore, each node, the *routing table* refers to, has to be located in a different cluster. As a result, the nodes preferred by proximity metric would be located close to the cluster boarder or at the threshold area. Due to this fact, *routing table* entries would faster become obsolete. As shown in Figure 13 the node that is two hops away from the *routing table* owner may be located geographically close to the cluster border. Therefore, this node may leave its cluster soon and, as a result, the *routing table* entries that refer to this node become obsolete.

Due to the drawbacks mentioned in the previous paragraphs, no proximity metric is used in the context of our Clustered Pastry MP2P system. Whenever multiple nodes are available that fulfill the requirements of the same table entry, the more recent link is preferred.

3.5.4 *Routing Algorithm*

The routing algorithm of Clustered Pastry is similar to that of the Pastry DHT. But as explained before they differ on the structure of the used routing tables. Clustered Pastry's *routing table* provides links to nodes that are located in other clusters and, therefore, this table can only be used for inter-cluster routing. When a lookup request has reached the cluster of the destination node, the *leaf set* is used for intra-cluster routing.

Based on these observations, the lookup mechanism works as follows: When a lookup is initiated, the local *leaf set* has to determine whether the requesting node or a virtual neighbor that is located in the same cluster is the destination of this lookup. If so, the lookup is completed or can be completed with a single overlay hop, respectively. Otherwise, the *routing table* is used to determine a node that matches at least a digit more of the destinations prefix. Thereafter, the request is forwarded. The next overlay hop node, which receives this request, also looks up its *leaf set* in order to determine whether the destination of the request can be found within its cluster. Otherwise, the request is, once again, forwarded to a node that matches at least one more prefix digit of the destinations overlay identifier. This is repeated until the request is received by the destination node.

However, as MP2P systems have to use a lossy wireless channel packets may get lost during transmission as a result of, e.g., collisions. As the UDP transport layer protocol does not support retransmissions, retransmissions have to be initiated and handled by the application itself. In order to increase the reliability of the lookup mechanism, acknowledge messages to the previous overlay hop are sent whenever a request message is received by a node. However, when an acknowledge message is not received within a specific time, it is assumed that the packet has been lost and the message is retransmitted. Yet, each node is only allowed to retransmit a request twice. Otherwise request messages would be flooded in the network whenever a node is temporary unavailable and, therefore, unable to reply an acknowledge message.

3.6 NODES AND OBJECTS

This section provides information on how nodes join and leave the network and mechanisms used to store objects in the network. Also the resulting maintenance of objects due to bootstrapping of nodes and churn is discussed.

3.6.1 *Storing, maintaining and Retrieving Objects in a Mobile Peer-to-Peer Network*

MP2P systems as considered in this thesis provide storage and retrieval services of data objects. In this subsection, we describe the mechanisms that are required to store, maintain, and retrieve objects in the network.

STORE AND RETRIEVE OBJECTS

In order to store an object in a Clustered Pastry MP2P network, the object overlay identifier has to be generated in a first step. After the generation of a new object identifier, a lookup for this novel object identifier is initiated by the node that provides this object in order to determine the root node (the node that is logically closest to the object). After receiving a reply message of the root, the object is transmitted and stored at the root node. According to the replication mechanism, multiple other steps may be necessary in order to distribute replicas in the network (as discussed in the next paragraph and as well in Chapter 6.1).

In order to retrieve an object, the overlay identifier of the object has to be determined by hashing the objects name. Based on this identifier a lookup for the root node is initiated. After receiving a reply from this node, the object can be downloaded directly.

BASIC REPLICATION

When objects are stored at a single node only, data may get lost when this node leaves the network spontaneously. Therefore, a basic replication mechanism is required to ensure that objects are available despite of node failure or unexpected leaving. As a result, objects are replicated on the two direct virtual neighbors of the root node. This approach is similar to the replication mechanism proposed by [89]. Yet, the distribution of those replicas is handled by the root node and not by the node that provides this data. As the root node and the roots of the replicas are within a single cluster, overhead generated due to the distribution of the replication is low.

Each node is able to detect when a node leaves or joins the cluster due to the periodically *leaf set* updates. When a node leaves a cluster, objects, and replicas that were stored at this node have to be redistributed in this cluster. However, it cannot be assumed that a node that leaves the network coordinates this redistribution of this objects. Therefore, the virtual neighbors of the node that has left the cluster are used to perform this task. In the first step, each of the virtual neighbors has to check the locally stored objects and replicas. On one hand, for each object stored at a virtual neighbor, a new replica has to be redistributed. On the other hand, objects have to be redistributed when a root node leaves the network. Whenever a node joins a cluster, the virtual neighbors have to check their storage for replicas and objects as well. When this new node is logically closer to an object or at least closer as the roots of the replicas, a copy of this object has to be stored at this new node.

When only few nodes are located within a cluster, the number of nodes may not suffice to distribute all replicas of an object. Therefore, objects have to be replicated to nodes in an adjacent cluster as well in such a scenario. Otherwise, objects may be lost when all nodes would leave a cluster. However, those externally stored replicas have to be stored within the cluster that is logically closest to the identifier of the object. Furthermore, also two replicas have to be stored in the adjacent cluster in order to ensure that these external replicas do not get lost due to nodes that leave the cluster.

Due to this replication mechanism it is ensured that no objects can get lost as a result of node departure. Security related challenges for replication mechanisms will further be discussed in Chapter 6.1.

3.6.2 *Joining and Leaving the Mobile Peer-to-Peer Network*

Whenever a node joins a DHT, the routing tables of this node as well as the tables of each virtual neighbor has to be updated. Furthermore, nodes that leave the MP2P network have to be handled by the network as well. In the following paragraphs bootstrapping mechanisms of Clustered Pastry and the effects of node departure are discussed.

BOOTSTRAPPING

When a node wants to join the network, a bootstrapping node is required. This bootstrapping node provides first *routing table* entries and further forwards the join request to the virtual neighbors of this new node. As a result of this procedure the new node is able to build its *routing table*. Furthermore, the routing tables of the bootstrapping node and the virtual neighbor is updated. So far, the bootstrapping mechanisms of the Clustered Pastry system is very similar to the mechanism used by

Pastry. However, Pastry requires a predefined (e.g., defined by the user) bootstrapping node while Clustered Pastry can harness the structure of the system in order to detect an adequate bootstrapping node. Furthermore, traffic due to bootstrapping can be reduced compared to the approach proposed by Pastry.

As the location aware approach combines the virtual with the geographical neighborhood, the geographical neighbors can be used to boot nodes. As each node in the network attach its node ID and *leaf set* information to the periodically broadcasted *hello* messages of the underlay, neighbors that participate in the DHT can be identified easily. As a result, the virtual neighbor of the new node can be reached within a single overlay hop and no costly routing process is required. Due to this fact, only little traffic overhead is introduced during bootstrapping.

LEAVING THE NETWORK

In the best case scenario, nodes that leave the network are able to notify the virtual neighbors and redistribute the locally stored objects in the cluster. However, this cannot be assumed as nodes may crash, lose the connection to the network or disconnect due to unexpected conditions from the MP2P network. As mentioned earlier, replicas are stored at the virtual neighbors and, therefore, leaving nodes will not result in lost objects. On the other hand, the previously proposed update mechanisms of routing tables ensure that the provided entries are fresh and that disconnected nodes are removed from the tables. Therefore, disconnecting nodes do not affect the MP2P system strongly but will only introduce a low traffic overhead.

3.7 COMPARISON OF LOCATION AWARE MOBILE PEER-TO-PEER SYSTEMS

Out of the approaches discussed in the related work (see also Chapter 2.1.4) the MADPastry [118] and PeerNet [34] MP2P systems are most closely related to our approach. Therefore, they are compared in detail to Clustered Pastry in this section.

3.7.1 *MADPastry*

MADPastry and Clustered Pastry are both based on the Pastry DHT. Both systems benefit from the location awareness of the peers. The geographical position of the nodes is used in order to optimize the routing algorithm of the MP2P systems. However, both systems differ in several ways.

The geographical groups formed by MADPastry are defined via a landmarking mechanism. Within each group, a node is chosen as landmarking node. This node is defined as the center of the cluster. Therefore, the geographical size and location of a cluster defined by MADPastry is dynamic and depends on the mobility of the landmarking node. As a result, the arrangements of these clusters cannot be influenced by the system but depends on the location of the landmarking nodes only. Furthermore, landmarking messages, which are required to assign nodes to their clusters, have to be broadcasted periodically in the cluster. Thus, overhead is generated. Moreover, those messages can be forged or manipulated by malicious nodes in order to attack the network. As most mobile devices provide reliable positioning system as GPS, landmarking mechanisms should be avoided.

Furthermore, MADPastry's *routing table* is truncated and provides only a single row. Therefore, only the first overlay hop is provided by this table. Also, the *leaf set* is strongly truncated and only small number of entries is maintained proactively. As a result, the number of required hops increases strongly in scenarios with an increased number of nodes.

In summary, MadPastry introduces benefits in scenarios where no deployment area can be defined due to the dynamic cluster location. Yet, due to the strongly truncated *routing tables* and the randomly distributed clusters in the deployment area, the efficiency of the lookup mechanism is limited.

3.7.2 *PeerNet*

PeerNet is based on a modern positioning mechanism. Each node in the network is assigned to the cluster in the network it is most likely to be encountered. For this reason, nodes do not adapt their identifiers when leaving an area, but provide information about their destination to their home cluster. Therefore, a proxy node is required at each cluster that stores this information and that is forwarding received requests to absent nodes when required. Those proxies have to be available all the time and are not allowed to leave the cluster.

However, those highly available entities that provide such services cannot always be assumed in the context of MP2P systems. Considering disaster relief scenarios, the nodes destination may further be hard to predict. Thus, either nodes have to report their current position to their proxy periodically or lookup messages may have to be rerouted multiple times following the trace of the nodes in the networks.

Finally, PeerNet is based on recent technology as GPS and uses a location aware routing algorithm. Yet, overlay identifiers are not adapted to the current location of the nodes, but are mapped on specific predefined home areas. Furthermore, PeerNet is based on assumptions as highly available proxies and the prediction of the nodes mobility that may not be valid in a disaster relief scenario. Our Clustered Pastry MP2P system also uses a GPS based location aware routing algorithm, but neither proxies nor a prediction of the participants mobility is required.

3.8 CHAPTER SUMMARY

In this chapter, the requirements of a communication infrastructure for a disaster relief scenario have been discussed. Based on the outcomes of this discussion, major challenges of this scenario as on demand usage or robustness have been outlined. As MP2P systems match those requirements they are highly recommended to be used as a communication substrate for disaster relief scenarios.

Therefore, our novel Clustered Pastry MP2P system has been introduced in the second part of this chapter. This system combines a MANET underlay with a structured P2P overlay. In order to match the challenges introduced by the scenario a location aware approach is used. Each node uses overlay identifiers that are adapted to the geographical location of the node. Due to this matching of the geographical position to the overlay identifier, a straight routing can be achieved. As a result of this geographically structured routing algorithm, the efficiency of the MP2Ps lookup mechanism can be strongly increased. Furthermore, a cross-layer is used to provide

underlay information to the overlay. Due to this information, the routing tables of the overlay can be maintained efficiently. Those routing tables have further been adapted in order to reduce update overhead and to meet challenges introduced by the scenarios.

After we have presented the Clustered Pastry MP2P system in this chapter, this system will be evaluated in the following chapter. Therefore, the influences of metrics, e.g., the number of participants and the mobility characteristics of the nodes have to be considered.

4

EVALUATION OF THE CLUSTERED PASTRY MOBILE PEER-TO-PEER SYSTEM

»It doesn't matter how beautiful your theory is, it doesn't matter how smart you are. If it doesn't agree with experiment, it's wrong«

— Richard P. Feynman

In the previous chapter we introduced Clustered Pastry, a MP2P system developed to operate in disaster relief scenarios. In this chapter a simulation based evaluation is conducted to prove the effectiveness of this Clustered Pastry system.

First, the used simulator, implementation details of Clustered Pastry and the default settings are briefly discussed. Thereafter, the evaluation of Clustered Pastry is provided based on multiple settings in order to identify the strength but also the limitations of our system.

4.1 EVALUATION SETTINGS AND METRICS

This section introduces the evaluation tools and settings that were harnessed to analyze the efficiency of our Clustered Pastry system. In the first part of this section, the goals and specific settings of the evaluation are discussed. Thereafter, we define a set of metrics that are used to evaluate the efficiency of our Clustered Pastry system. Furthermore, we introduce the evaluation methods as well as the default settings and parameters.

4.1.1 *Goals of the Evaluation*

In the following paragraphs, we define the goals for the evaluation of our Clustered Pastry system. We distinguish between three major goals: the efficiency of Clustered Pastry compared to other MP2P systems, the evaluation of the optimization mechanisms that have been developed for the Clustered Pastry system, and the analysis of influences introduced by the disaster relief scenario.

COMPARISON BETWEEN CLUSTERED PASTRY AND A LAYERED APPROACH

The first goal is to show the applicability of our Clustered Pastry system in a disaster relief scenario. Therefore, we compare our system to a reference MP2P system. This reference system is based on a layered MP2P architecture and, therefore, does not benefit from cross-layer information, but uses the same underlay and overlay protocols as Clustered Pastry. Due to this fact, an optimal comparability between the reference MP2P systems and Clustered Pastry is achieved.

OPTIMIZING CLUSTERED PASTRY

Beyond comparing our system to other MP2P systems, our second goal is to validate

the mechanisms that have been proposed in the previous chapter to stabilize and optimize Clustered Pastry. Therefore, we will evaluate the impact of threshold areas and determine the optimal size of those areas. Moreover, traffic reduction mechanisms as proposed in Chapter 3.5.3 are analyzed.

In Chapter 3.4.2, threshold areas have been introduced. The influence of those areas on the stability of the network and the traffic generated due to nodes that leave a cluster has to be analyzed. As a result, the efficiency of this threshold area as well as a best effort parameter for the size of these areas has to be determined.

Moreover, we introduced four mechanisms in the previous chapter that can be used to reduce the traffic that is generated by the *leaf set* update mechanism. The efficiency of those mechanisms and the influence on the Clustered Pastry systems will be evaluated in the remainder of this chapter.

INFLUENCES OF THE DISASTER RELIEF SCENARIO ON CLUSTERED PASTRY

The third goal addresses influences of the disaster relief scenario. In particular, the number of network participants, the node mobility, and the background traffic has to be analyzed.

As defined by our scenario, we have to assume up to two hundred nodes that participate in a disaster relief operation. Therefore, the scalability of the Clustered Pastry system in settings considering the scenario specific amount of participants has to be evaluated.

MP2P systems consist of mobile devices. Yet, the mobility of units in a disaster relief scenario is not random but depends on a large list of characteristics including the position and the duties of the specific unit. In order to ensure that Clustered Pastry is able to provide services in a disaster relief scenario, it is essential to evaluate our approach based on a realistic mobility model.

Beside the storage and retrieval services, other services may be provided by the MANET substrate in a disaster relief scenario. As a result, background traffic may be introduced by, e.g., voice communication. Therefore, we have to analyze the effects of background traffic on the efficiency of our Clustered Pastry system.

4.1.2 *Evaluation Metrics*

It is essential to define the primary metrics that are used to analyze the results of the simulations. Those metrics are used to determine the efficiency of the Clustered Pastry MP2P system and to analyze the impact of the variation of parameters on the reliability of this system. In the context of this evaluation we are using two basic metrics, which are typically used in the related work to determine the efficiency of an MP2P system (e.g. [118][29][18]). However, additional metrics are required to evaluate specific settings. Those metrics will be introduced in the corresponding sections.

As the availability of the services of an MP2P network in a disaster relief scenario is essential, the fraction of failed lookups (f_{lookup}) is used to determine the reliability of our Clustered Pastry systems. This metric determines how reliable an MP2P network operates regarding the retrieval and availability of data objects and, for this reason, describes the usability of our Clustered Pastry system. This metric is determined by comparing the number of initiated lookups with those that are completed successfully.

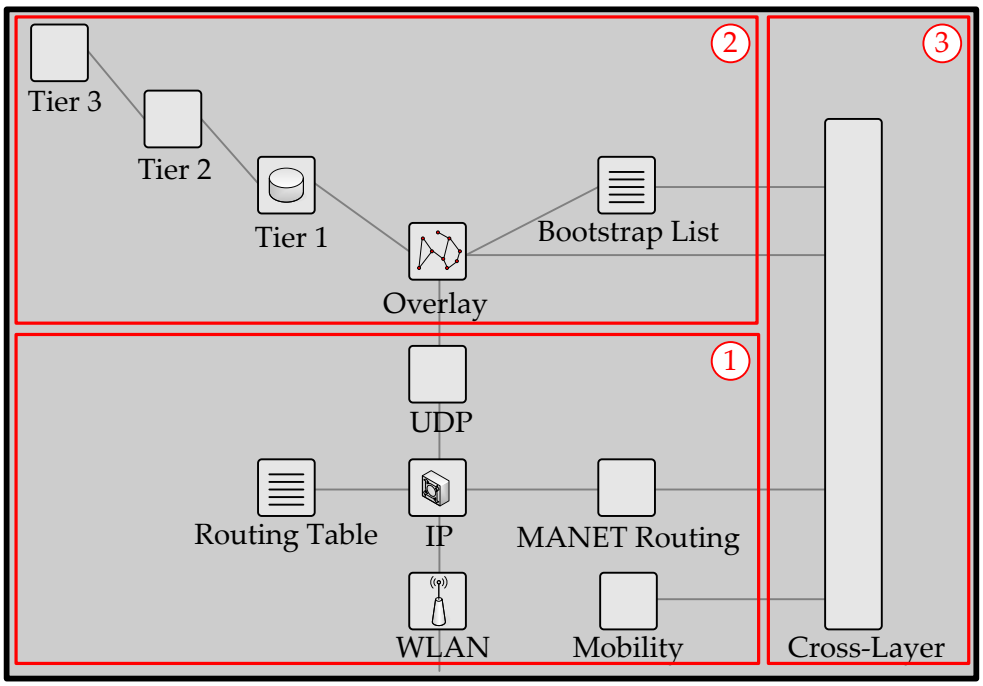


Figure 14: Structure of the OMNeT implementation of Clustered Pastry

As mentioned before, bandwidth is strongly limited. Therefore, the traffic that is introduced by the system (T) may indicate the efficiency of our Clustered Pastry system. We distinguish four traffic metrics in the context of this thesis.

The overlay traffic ($T_{overlay}$) includes traffic generated by updating the *routing table* as well as traffic due to storing, retrieving, and maintaining objects in the network. Yet, traffic introduced due to cross-layering is neglected. The cross-layer traffic (T_{cross}) on the other hand displays traffic generated by the *leaf set* update mechanism and the *routing table* updates provided by cross-layer information. The overall traffic ($T_{overall}$) combines the overlay ($T_{overlay}$) with the cross-layer traffic (T_{cross}) and, therefore, displays the total traffic that is introduced by the Clustered Pastry system to the network. Yet, traffic generated by the OLSR ETX protocol itself is excluded. The previously mentioned traffic metrics display different traffic characteristics of the Clustered Pastry system in terms of transmitted data volumes. The fourth traffic metric introduces the average number of underlay hops that are required to store or lookup a message in the network (T_{hops}). While the previous metrics indicate overhead generated by the Clustered Pastry system, this last traffic metric is used to show the efficiency of the geographical clustering in terms of the number of average underlay hops.

4.1.3 *Simulator Selection*

A simulation enables the evaluation of complex systems with reasonable costs. Moreover, we are able to analyze the influences on the MP2P system in detail. A testbed-based evaluation on the other hand is limited to either a small number of overall nodes or a static setting of these nodes. Furthermore, background influences that affect such an testbed a very hard to predict. Therefore, we validate the efficiency of

our Clustered Pastry system by the means of simulation. However, the simulation infrastructure that is harnessed for our evaluation has to be selected carefully. This network simulator must be extensible and modular.

As a result of these requirements, the modular discrete event simulator OMNeT++ [111] is used to evaluate our Clustered Pastry approach. This object-oriented open source simulator is well known and enables the simulation of a large set of communication networks. The protocols, applications and components that have to be simulated are implemented in C++. Network specifications and the interaction of those protocols, applications, and components are defined via the Network Description language. Based on this simulation infrastructure, several frameworks have been developed that enable the simulation of a wide range of different networks. Due to this fact, frameworks based on the OMNeT++ simulator are used by both the MANET as well as the P2P research community. Two of those frameworks are used in our evaluation.

On one hand, we used the INET framework [4]. This framework provides implementations of MANET routing protocols including OLSR with and without the ETX extension. Furthermore, basic applications such as simple traffic generators, mobility models, models for wireless channels, transport layer, network layer, and link layer protocols are included. Therefore, the basic underlay protocols that are required to simulate our Clustered Pastry system can be provided by this framework.

On the other hand, our Clustered Pastry model is based on the OverSim [11] framework. This framework is a collection of structured P2P systems and provides implementations of P2P systems including, e.g., Chord, Pastry, and CAN. Also traffic generators for P2P scenarios are introduced by this framework. Furthermore, abstract models to simulate the behavior of wired underlay networks are available. However, no mobile or wireless underlay is provided by the OverSim framework.

In order to implement the Clustered Pastry system, we had to combine the INET with the OverSim framework. The resulting structure of protocols, applications, and components that are used to simulate a Clustered Pastry node is shown in Figure 14.

The underlay (1) consists of an 802.11g WLAN interface, which includes physical and link layer protocols. Furthermore, IP is used as network layer protocol. Also the routing protocols of the MANET operate at the network layer. We adapted the OLSR MANET protocol to attach update information of the overlay's routing tables to the periodically sent *hello* messages. The transport layer is the uppermost layer of the underlay and uses UDP.

The DHT itself is implemented in the overlay (2). This includes the routing algorithm of the structured P2P system as well as the routing tables. This overlay is further linked with three optional tiers of applications. The application on the first tier is used to store and maintain objects and replicas during our simulations. The second tier implements a traffic generator. This traffic generator periodically triggers the storage and lookup of objects in the MP2P system. The third and final tier is not used during our simulations. We modified the routing algorithm, the structure and the update mechanism of the routing tables, and the allocation of overlay identifiers of the Pastry implementation provided by OverSim as discussed in the previous Chapter.

In order to enable functionalities as the *leaf set* updates via the underlay's *hello* messages, we implemented a cross layer (3) between the overlay and the MANET routing mechanism. This cross-layer is also linked to the mobility component in order to enable the location awareness and to the list of bootstrapping nodes to provide

geographically close bootstrapping nodes. All other components are required either by the underlay or overlay implementation.

4.1.4 *Default Simulation Parameters*

Next, we define the settings of Clustered Pastry's parameters as used for the simulation based evaluation. Those default settings are used for each of the following settings, except where otherwise specified.

The Clustered Pastry's underlay is based on an OLSR ETX MANET protocol as described in Section 3. Default parameters as defined by the OLSR Request for Comments (RFC) 3626 [21] have been used during the evaluation. Furthermore, each node uses a WLAN interface according to the 802.11g standard [3] and we assume an average transmission range of about 200m.

The overlay lookup functionality is provided by the Clustered Pastry DHT. As discussed in Chapter 3.5.1, the parameter b is set to 1, which results in two clusters per level. The cluster *level*, on the other hand, is per default 2. Due to this fact, settings with four clusters were simulated during our evaluation. Up to 50 objects are provided by the nodes within the DHT. Those objects are distributed equally in the namespace. Each object has a size of 2kbytes and a lifetime of 250s. Outdated objects are discarded and replaced by a new one. Two replicas of each object are generated and stored by the virtual neighbors as discussed in Chapter 3.6.1. After the booting phase, a lookup for a randomly selected object is initiated every ten seconds on average.

The deployment area is defined as the field, all first respond units are located at. Thus, all nodes operate within the boundaries of this area. As a dense network is assumed due to the scenario, a field size is chosen where, a connected network is typically achieved, i.e., a route between an arbitrarily chosen pair of nodes can be established with a high probability. Therefore, the Ad-hoc Network Simulator (ANSim) [50] has been used to determine the size of the deployment area. This tool simulates node mobility and derives the connectivity between nodes within the network as a function of the node density and the average transmission range. The resulting area of operation derived by ANSim when assuming a connection probability of about 99% is shown in Table 5. Furthermore, the node mobility is defined by a random waypoint model. This model has been implemented as part of the INET framework. Each node moves with a maximum speed of 1m/s and changes its speed and direction after a time period between 10s and 60s. Yet, in Chapter 4.6 a more realistic mobility model for disaster relief scenarios [110] is used. Each simulation runs for 30 simulated minutes. To indicate the deviations of the results of the different simulation runs, confidence interval are displayed with a level of significance of 95%.

An overview of the used parameters is shown in Table 3. More details on the simulators default settings have been attached to Appendix A.1.

4.2 COMPARING CLUSTERED PASTRY WITH A LAYERED APPROACH

In the previous chapter, our Clustered Pastry system has been discussed in depth. Multiple modifications and adaptations of the underlying protocols have been introduced in order to enable this system to operate efficiently in a decentralized and

Nodes	Deployment area
25	500m x 500m
50	800m x 800m
75	900m x 900m
100	1100m x 1100m
150	1400m x 1400m
200	1550m x 1550m

Table 2: Number of nodes and deployment area

Basic parameters	
Nodes	25 - 200 (default: 100)
Mobility model	Random waypoint
Node speed	0-1 m/s
Initial node placement	Equally distributed
Simulation time	30 minutes
Overlay parameters	
Basic protocol	Pastry
Routing algorithm	semi-recursive
Leaf set size	dynamic
ID size	32 bit
Parameter b	1
Request frequency	10 s (global)
Currently stored objects	50
Life time of Objects	250 s
Object size	2 kbyte
Underlay parameters	
Basic protocol	OLSR
Routing metric	ETX
Transmission range	200m
Frequency *hello* messages	2 s
MP2P parameters	
Threshold area	40m
Number of clusters	2 - 16 (default: 4)
Leaf set update frequency	2 s
Routing table update frequency	10 s
Lookup retransmissions	2

Table 3: Overview of parameters used in the simulations

	Basic layered	*Adapted layered*	Clustered Pastry
Size of *leaf set*	16	4	Nodes in Cluster
Routing table Rows	Undefined	Undefined	2
Size of Identifier	128 bit	32 bit	32 bit
Parameter b	4	2	1
Clusters (c)	1	1	2/4

Table 4: Settings of the different evaluated Mobile Peer-to-Peer systems

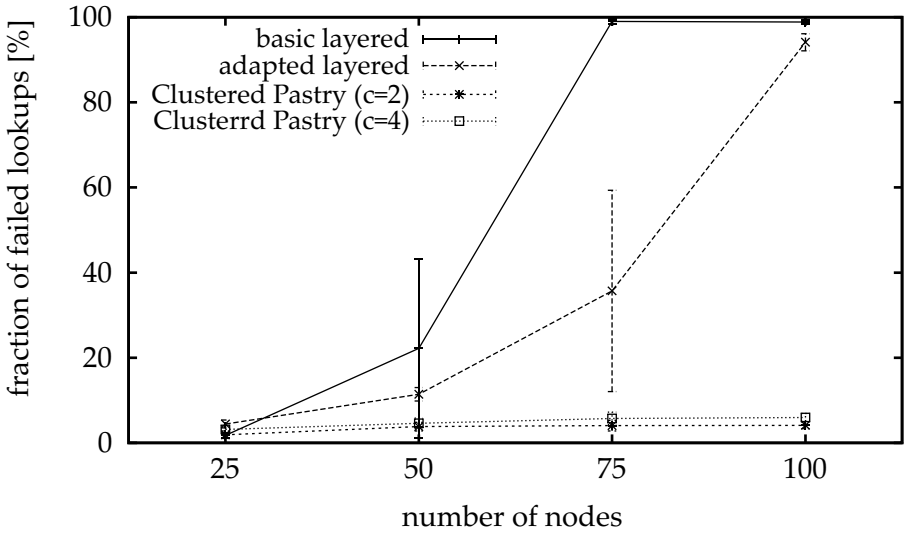


Figure 15: Fraction of failed lookups in mobile Clustered Pastry and non-clustered Pastry scenarios

mobile environment. In this section the Clustered Pastry system is compared with a layered reference MP2P system in order to show the shortcomings of layered systems and to highlight the benefits of Clustered Pastry as defined in our first goal. This layered system consists of an OLSR based underlay that uses the ETX metric combined with a Pastry overlay network. The layered approach is evaluated based on two different sets of parameters. The first set uses the default parameters provided by the DHT's specifications and is hereinafter referred to as the *basic layered setting*. The second set is adapted to the mobile, wireless scenario. Therefore, the *leaf set* size, the parameter b, and the size of the overlay identifiers have been reduced. All these modifications result in a reduction of the traffic introduced by the update mechanisms of Pastry's routing tables. This second set is hereinafter referred to as the *adapted layered setting*. An overview of the overlay settings is shown in Table 4.

The evaluation of these scenarios is based on three basic metrics. As the availability of the services provided by the MP2P system is essential, the fraction of failed lookups (f_{lookup}) is used to determine the reliability of the MP2P systems. Furthermore, the traffic that is introduced by the systems may indicate the efficiency of these approaches. Therefore both, the average number of sent messages (T_{hops}) and the resulting overall traffic ($T_{overall}$) are monitored.

4.2.1 *Results of the Comparison of the Mobile Peer-to-Peer Systems*

In the following paragraphs, the evaluation results of the Clustered Pastry system and the reference model are discussed in detail. Based on the three previously mentioned metrics, the benefits and flaws of both approaches are discussed.

FRACTION OF FAILED LOOKUPS

As the availability of the lookup and retrieval services is essential, the fraction of failed lookups is considered as the most important metric. The reference model based on the previously introduced settings and the Clustered Pastry systems provide very good results in small scale scenarios with 25 nodes, as shown in Figure 15. Only few lookups are dropped in these scenarios, though the number of failed lookups based on a *adapted layered setting* is twice as high compared to the Clustered Pastry and the reference model with the *basic layered setting*. However, considering settings with an increased number of nodes, the fraction of failed lookups is highly increased when using the reference model. Even though the *adapted layered settings* provide better results than the *basic layered settings* when considering more than 50 participating nodes in the network, the fraction of failed lookups is unacceptably high. Though, the cross-layered Clustered Pastry system provides reliable lookup operations even beyond this number of participants.

Both the excellent results in small scale scenarios and the reduced availability in settings with an increased number of nodes is a result of the structure of the routing tables of the reference model and the resulting traffic generated during bootstrapping. However, our Clustered Pastry system provides reliable results due to the location aware lookup mechanism, the structure of routing tables, and update mechanisms.

TRAFFIC

Considering the *basic layered setting* of the reference model, 16 links to virtual neighbors are provided by the *leaf set* and at least 15 links to nodes that are logically distant are stored at the *routing table*. Due to this fact, most of the 25 nodes in small scale scenarios can be reached within a single overlay hop. As only few peers are participating and as no churn is simulated in this scenario, only few overhead is introduced due to the building of these tables. However, the high amount of links that has to be provided by the routing tables introduces a major drawback when increasing the number of nodes. On one hand, the *routing table* has to provide multiple rows and, therefore, a higher amount of entries when the number of nodes is increased. On the other hand, the *leaf set* has to be updated whenever a node in the virtual neighborhood joins the network. As a result, most nodes are not able to build their routing tables due to missing entries before a timeout occurs. Therefore, these nodes are not able to bootstrap but initiate a new bootstrapping a few seconds later. As a result, only 85% of all nodes in settings with 50 nodes are able to boot when using a the reference MP2P system based on a the *basic layered settings*. Even worse, when increasing the number of nodes to 75, less than half of the nodes are able to successfully complete the bootstrapping and in scenarios with 100 nodes even less nodes are able to join the network. This results in a low availability of the stored objects in settings with the reference MP2P system based on the *basic layered settings* as nodes and, therefore, objects stored at these nodes are only temporary available. This can also be seen when considering the overall traffic

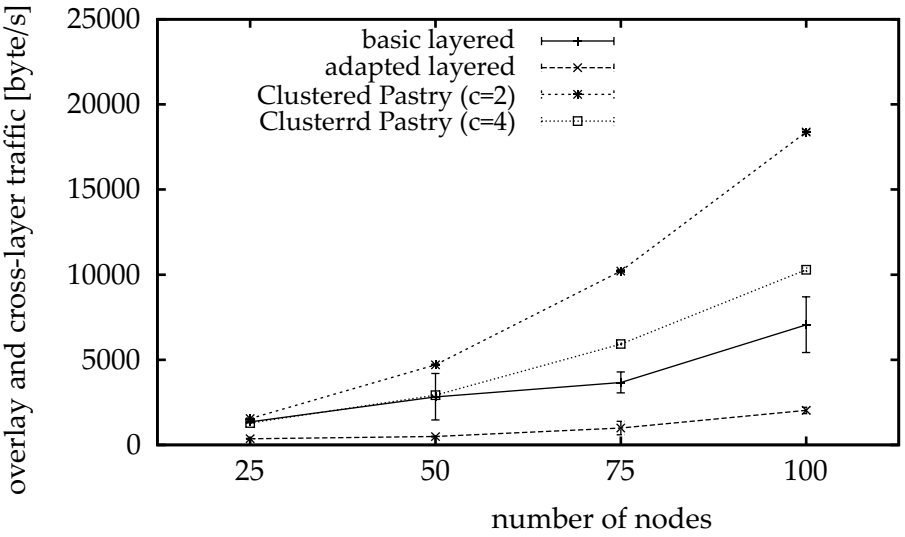


Figure 16: Traffic induced by mobile Clustered Pastry and non-clustered Pastry systems

as shown in Figure 16 and, in particular, the overall number of messages as shown in Figure 17 is strongly increased in large scale scenarios.

The reference model based on the *basic layered setting* performs worse regarding the fraction of failed lookups in small scale scenarios, as fewer links are stored in the routing tables and, therefore, most nodes cannot be reached within a single hop. Yet, also less update messages are required to maintain the routing tables. Due to this fact, this approach is able to perform better than the unadapted approach in scenarios with an increased number of participating nodes. Yet, the results are also strongly degrading when the number of nodes is increased beyond 50 nodes due to the same reason. However, as shown in Figure 16, the overall traffic introduced by this approach is very low compared to the Clustered Pastry system.

Despite the better results of in terms of a reduced fraction of failed lookups, traffic introduced by the Clustered Pastry is higher compared to the reference model as a result of the update mechanisms and the cluster based churn as discussed in the previous chapter. Yet, messages are mostly being sent to geographical neighbors and, therefore, can be transmitted more efficient in terms of a higher success ratio and a reduced traffic overhead.

4.2.2 *Summary of the Comparison of the Mobile Peer-to-Peer Systems*

To sum it up, the reference model provides reliable results in small scale systems due to the structure of the routing tables. Even though a higher traffic is introduced by the Clustered Pastry system as a result of the periodic routing updates, more stable results can be achieved. Due to this fact, we have shown that a cross-layered, cluster based system is more efficient for mobile, decentralized scenarios than a layered approach and, therefore, meet the first goal of this evaluation.

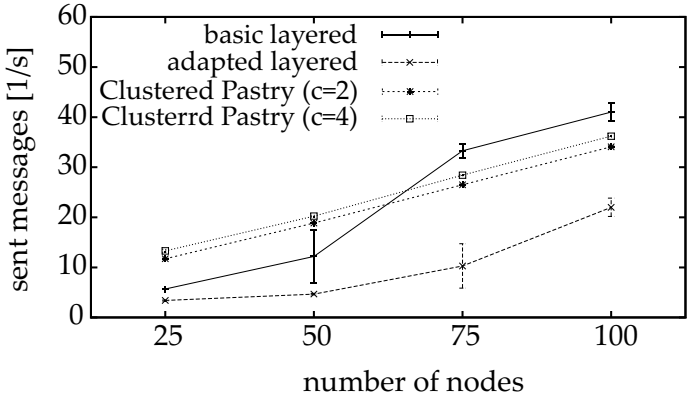


Figure 17: Average number of messages sent by mobile Clustered Pastry and non-clustered Pastry systems

4.3 EVALUATION OF THE THRESHOLD AREAS

To further investigate our Clustered Pastry system and the parameters, the threshold areas as introduced in Chapter 3.4.2 are analyzed in detail. The evaluation of the threshold area is further a requirement introduced by our second goal.

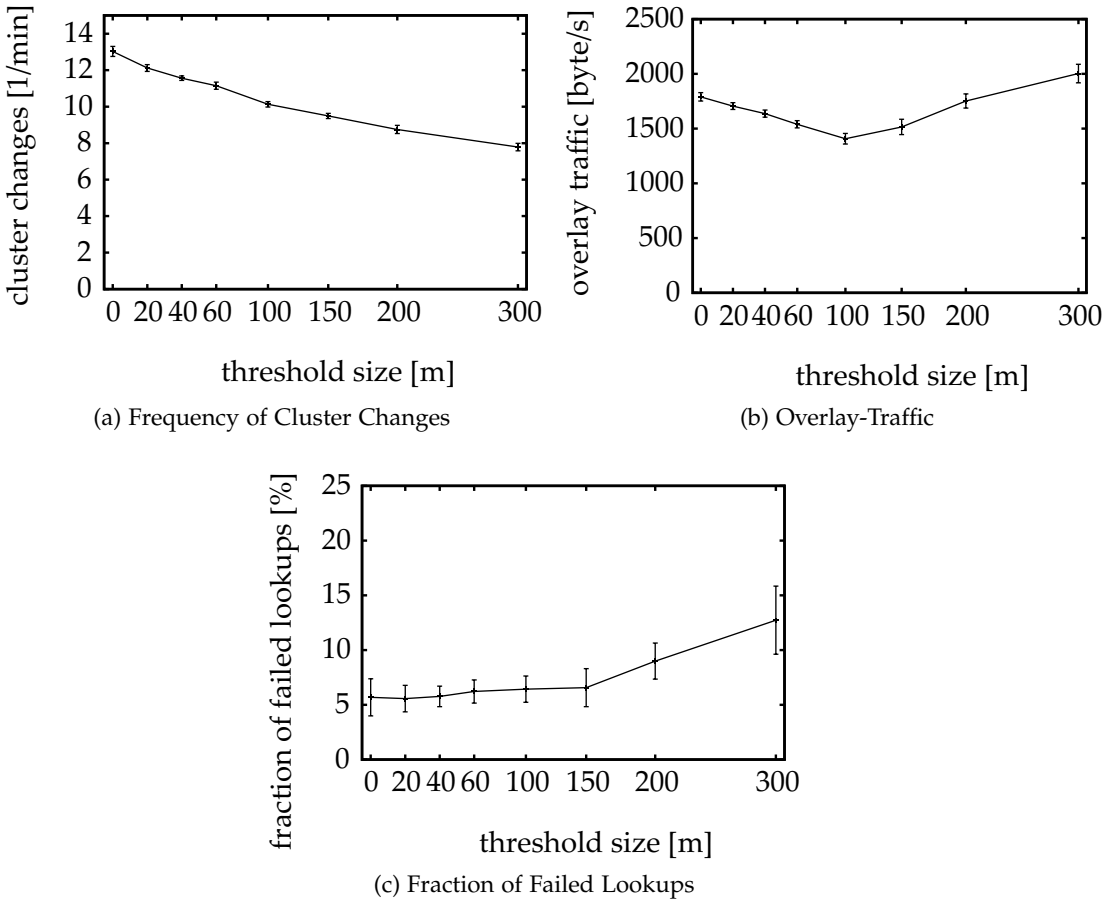
Whenever a node leaves or joins a cluster, all routing tables maintained by this node itself and all routing tables that provides links to this node require an update. As these updates result in traffic overhead, the number of cluster changes has to be minimized. Threshold areas as introduced by Clustered Pastry may be used to reduce the frequency of cluster changes in the network. However, those threshold areas, which are used as a buffer between the clusters, may affect other characteristics of our Clustered Pastry system as well.

In the following paragraphs, influences of the threshold areas on the Clustered Pastry system are evaluated. Therefore, eight settings are considered that differ in the size of the used threshold areas. In the first scenario, the size of the threshold area is reduced to zero and, therefore, no threshold area has been used at all. In the subsequent scenarios the size of the threshold area has been increased up to 300m. For all other parameters, default settings as described in Chapter 4.1.4 were used.

The efficiency of the threshold areas is evaluated based on three parameters. As the threshold areas were developed to reduce the overall number of nodes that leave a cluster, the average number of nodes that leave a cluster is analyzed ($f_{cluster}$). This frequency affects the availability of nodes as well as the traffic generated by the overlay network. This traffic introduced by the overlay ($T_{overlay}$) is analyzed in detail as well. Traffic due to underlay and cross-layer communication is neglected in this section, as this traffic is not affected due to cluster changes. Furthermore, the fraction of failed lookups (f_{lookup}) is harnessed to determine the availability of the services provided by the network.

4.3.1 *Results of the Evaluation of the Threshold Areas*

The threshold areas have been developed in order to stabilize the network and to



(a) Frequency of Cluster Changes

(b) Overlay-Traffic

(c) Fraction of Failed Lookups

Figure 18: Effects of threshold areas on the Clustered Pastry system

reduce the traffic introduced by cluster based churn. In the following paragraphs we evaluate those threshold areas and their impact on the Clustered Pastry system.

FREQUENCY OF CLUSTER CHANGES

Due to the threshold area, nodes have to adapt their own identifier only when they are passing the cluster border and the proper threshold area. As a result, the frequency of cluster changes can be affected by the size of this area as shown in Figure 18a. The number of nodes that leaves a cluster can be reduced by more than 10% by introducing a threshold size of 40m.

TRAFFIC

This reduced frequency affects, moreover, the traffic generated by the overlay network. As mentioned before, traffic is introduced whenever a node joins or leaves a cluster. This overhead includes traffic that is generated by the redistribution of objects as well as due to updating routing tables. When reducing the number of nodes that have to change their cluster, this overlay traffic is reduced in an equal proportion as shown in Figure 18b.

FRACTION OF FAILED LOOKUPS

As previously mentioned, the availability of nodes is influenced by the frequency of

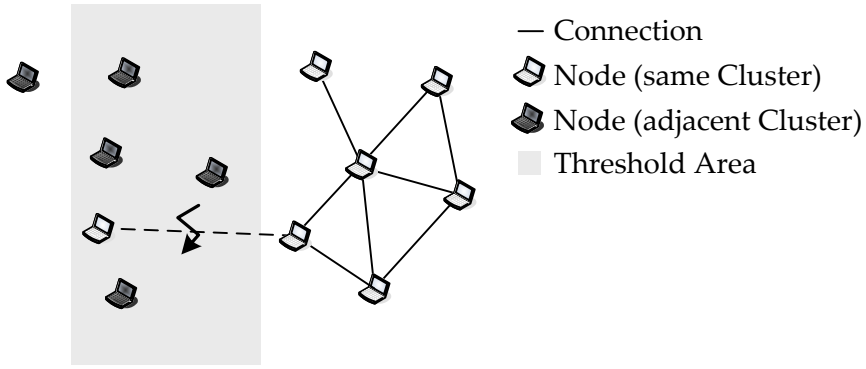


Figure 19: Limitations of the threshold area

cluster changes as well. Nodes are unable to provide services for a short amount of time while refreshing their routing tables after changing the cluster. However, those updates can be performed efficiently due to the characteristics of the Clustered Pastry system. Due to this fact, nodes are unavailable for about 0.95% of the overall simulation time on average when considering settings without threshold areas. However, this downtime can be reduced to 0.73% when using threshold areas with a size of at least 40m.

On the downside, large threshold areas affect negatively the fraction of dropped requests. As shown in Figure 18c, the fraction of dropped lookups increases in scenarios using a large threshold area. As a result of increasing this area, nodes may diverge from its own cluster without being forced to join another cluster. Thus, a node may be disconnected from its own cluster as shown in Figure 19. This results in stale routing tables, as no update messages can be received anymore. However, the fraction of dropped lookups is only affected in scenarios with a threshold size that exceeds 60m.

4.3.2 *Summary of the Influence of Threshold Areas on the Clustered Pastry System*

The threshold area is able to reduce the traffic introduced by the overlay network by reducing the frequency of the cluster changes. Yet, large scale threshold areas further affect the fraction of dropped lookups and, therefore, the availability of the services provided by the Clustered Pastry system negatively. Even though settings with a threshold size of 60m up to 150m provide still good results, single simulation runs have shown that these settings may affect the reliability of the lookup mechanism negatively. This may results in an increased fraction of failed lookups. Thus, we use a threshold area with a size of 40m for our disaster relief scenarios in this thesis. Thus, we were able to satisfy a requirement introduced by our second evaluation goal as we have shown that threshold areas are able to reduce the overlay traffic and we were able to provide a default parameter for the size of the threshold areas. However, in settings that either differ strongly on the transmission range or the node's speed, it may be necessary to adapt the size of the threshold areas.

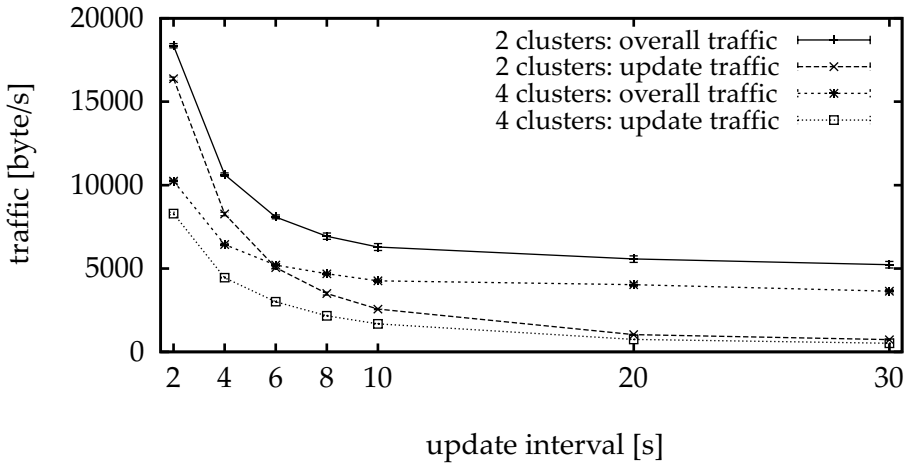


Figure 20: Evaluation results of the adapted *leaf set* update frequency

4.4 REDUCING THE TRAFFIC GENERATED BY THE LEAF SET UPDATE MECHANISM

The periodic updates of the *leaf set* are necessary to ensure fresh links to virtual neighbors, but also generate a high amount of traffic as mentioned in the previous chapter. About 80% of the traffic introduced due to the maintenance of the Clustered Pastry system is generated by this update mechanism (considering a setting with 100 nodes and 4 clusters). Thus, reducing the size of the update messages or the update frequency could strongly reduce the overall traffic load and, therefore, has been defined as part of the second evaluation goal. However, reducing the traffic should not be achieved at the expense of the freshness of the *leaf set* entries.

Three different approaches have been proposed in Chapter 3.5.3 to reduce the traffic generated by the *leaf sets* update mechanism. In the following paragraphs, the impact of those mechanisms is evaluated. Therefore, two metrics are used to determine the efficiency of these approaches. Obviously, reducing traffic introduced by the *leaf set* update mechanism affects the *leaf set* itself. As a faulty routing table has an impact on the reliability of the routing algorithm, the fraction of failed lookups (f_{lookup}) is used to determine the efficiency of this algorithm and the availability of the networks services. The second metric that is used to evaluate the efficiency of the proposed mechanisms is the traffic (T) generated by the Clustered Pastry system. This includes the overall traffic ($T_{overall}$) and, in particular, the cross-layer traffic (T_{cross}) introduced by the *leaf sets* update mechanism.

4.4.1 *Adapting the Leaf Set's Update Frequency*

As mentioned before, adapting the update frequency is a straight forward approach to reduce the traffic generated by the *leaf sets* update mechanism. As those update messages are coupled to the periodic *hello* messages of the OLSR protocol in the underlay, the frequency cannot be chosen freely. These *hello* messages are sent every two seconds to the geographical neighbors. Thus, the update frequency can be adapted by attaching the *leaf set* update information to every f_{LS_update}th *hello* message only.

As a result, the traffic generated by the update mechanism should be reduced by the factor f_{LS_update}. However, each node still attaches its own overlay identifier to each of the *hello* messages.

In the following evaluation, the f_{L_update} is increased from 1 to 15 resulting in update intervals from 2s to 30s. Furthermore, settings with two and four clusters are simulated. All other parameters are set to their default values as defined in Section 4.1.4.

4.4.2 *Results of the Evaluation of the Adapted Leaf Set Update Frequency*

In the following paragraphs, we discuss the evaluation results of the first traffic reduction mechanism that has been developed in the scope of this thesis to reduce the traffic generated by the *leaf set* update mechanism.

TRAFFIC

As shown in Figure 20, the traffic introduced by the *leaf sets* update mechanism can be reduced by increasing the update frequency as assumed previously. As a matter of course, this affects as well the overall traffic of our Clustered Pastry system. Though, while the traffic generated by the update mechanism strongly decreases when f_{LS_update} is increased, the overall traffic does not decrease in the same pace. This is on one hand a result of the increased number of required *routing table* updates. As mentioned in Chapter 3.5.2, *routing table* updates are also propagated via the data sets of *hello* messages for a strongly limited amount of time. Due to this fact, reducing f_{LS_update} also affects the freshness of the *routing table* entries. As a result, *routing table* updates are more often triggered by the Clustered Pastry system. Furthermore, rerouting of misrouted messages also introduces traffic overhead. For example, assuming that the *leaf set* of a node is stale and a received lookup message is not forwarded correctly to the destination node, but to a node that has recently left the cluster. Mostly, this does not result in a failed lookup. It is more probable that this 'stray' lookup message is once again forwarded to its destination cluster. Due to these effects, increasing the update interval beyond 6s does not provide any reasonable result regarding the reduction of the overall traffic. Only a marginal reduction of the traffic can be achieved beyond this point, especially considering scenarios with two clusters.

FRACTION OF FAILED LOOKUPS

However, while increasing the update interval, the average fraction of failed lookups is mostly stable until reaching a f_{LS_update} of 3 (an update interval of 6s). Beyond this point, the fraction of failed lookup increases due to the high fraction of stale routing tables as shown in Figure 21. These results are analogous in simulations with two and four clusters. However, the efficiency of scenarios with a higher number of clusters is more stable when increasing f_{LS_update}. As each node attaches its own overlay identifier to every sent *hello* message, fresh routing table entries to geographical neighbors are ensured. As a result, the *leaf set* of small scale clusters provides more reliable links as most of those linked nodes are as well geographical neighbors.

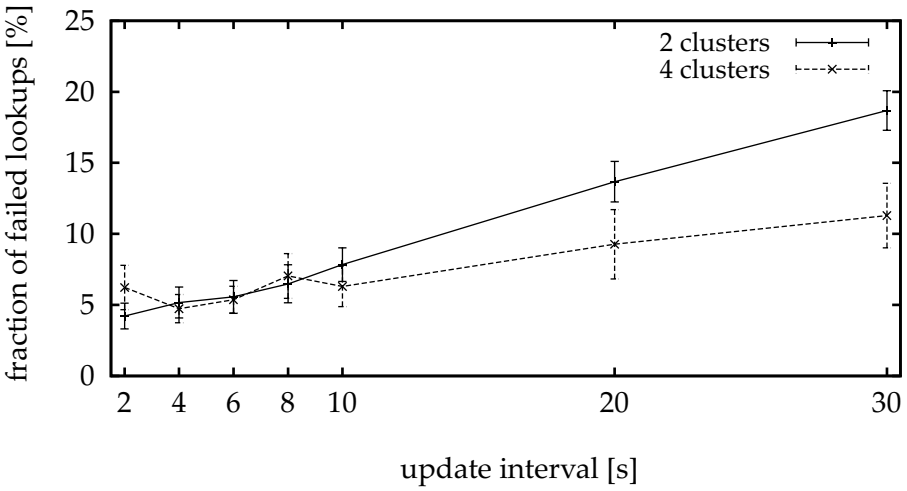


Figure 21: Evaluation results of the adapted *leaf set* update frequency

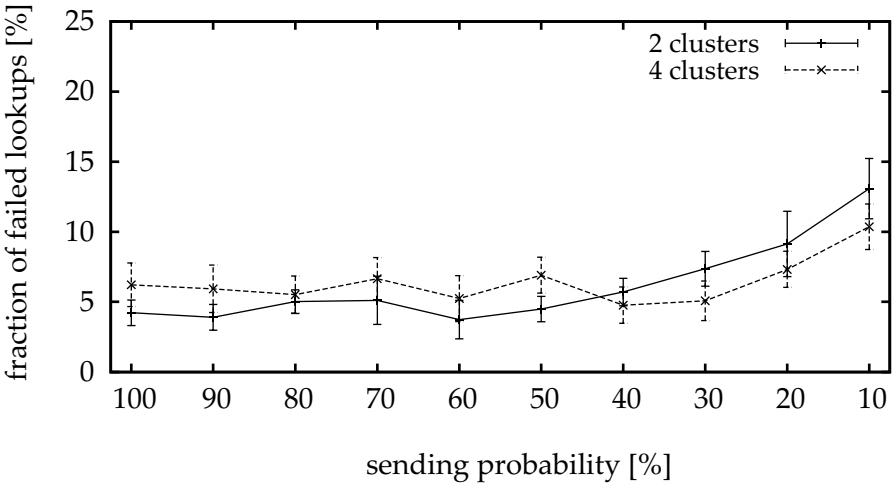


Figure 22: Evaluation results of the adapted *leaf set* gossiping mechanism

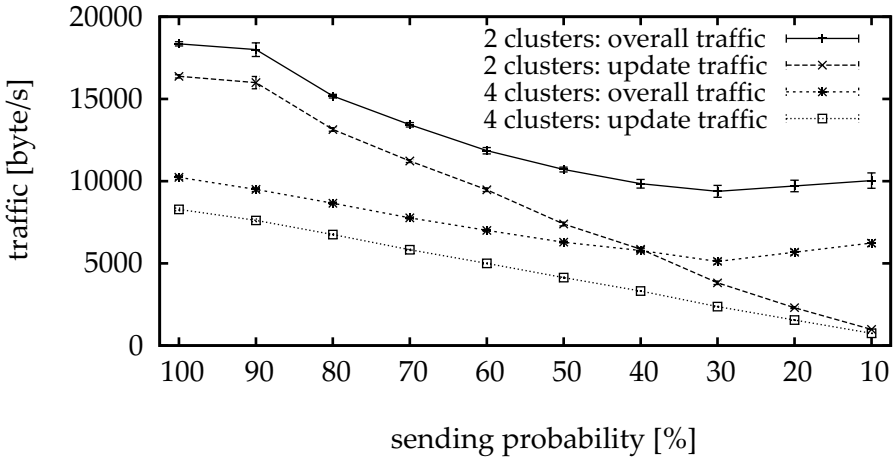


Figure 23: Evaluation results of the adapted *leaf set* gossiping mechanism

4.4.3 *Gossiping Mechanism for a Dynamic Number of Data Sets*

Gossiping in terms of sending data only with a specific probability is wide spread in wireless networks such as MANETs. Due to this fact, it is reasonable to harness gossiping in order to reduce the traffic introduced by the update mechanism of Clustered Pastry's *leaf set*. We have discussed different gossiping based approaches in the previous chapter that have to be evaluated in the following paragraphs.

The first approach attaches each data set only with a defined probability p_{gossip}. By reducing this probability, the overall traffic can be reduced. As the same update messages are sent by each node in the cluster, the probability is quite high that a node is able to extract the complete update information by accumulating the received data sets.

In order to evaluate this first gossiping approach, the probability p_{gossip} is varied from 100% down to 10%. All other parameters are set to their default values as defined in Section 4.1.4.

4.4.4 *Results the Gossiping Mechanisms Based on a Dynamic Number of Data Sets*

In the following paragraphs, we discuss the results of the evaluation of a gossiping mechanism that is used to reduce traffic generated by the *leaf set* update mechanism. This first gossiping approach is based on a fixed probability p_{gossip} and a dynamic number of data sets per *hello* message.

TRAFFIC

In Figure 23, the overall traffic and the update traffic of settings with 2 and 4 clusters are displayed as a function of the sending probability of the data sets (p_{gossip}). It is shown that the reduction of the gossiping probability p_{gossip} directly affects the cross-layer traffic. The overall traffic can also be reduced by means of gossiping. Yet, reducing p_{gossip} below 30% results in an increased overall traffic due to the affects of stale routing tables as discussed in the previous section.

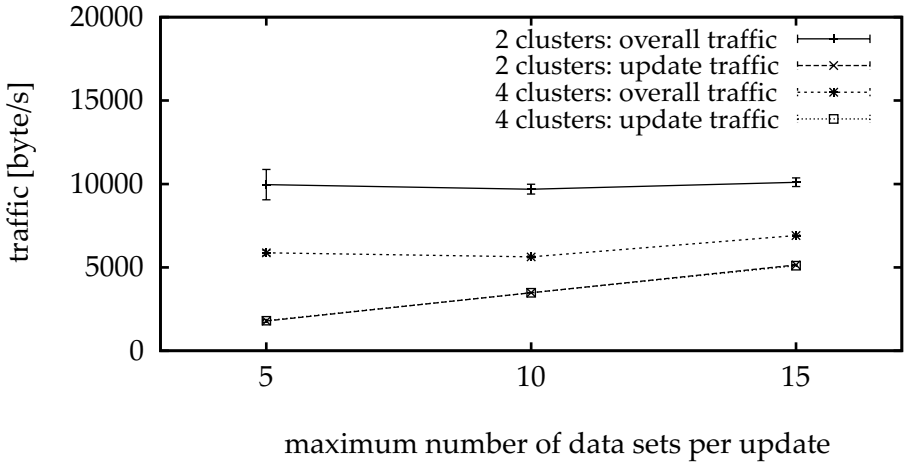


Figure 24: Evaluation results of the adapted *leaf set* gossiping mechanism

FRACTION OF FAILED LOOKUPS

The fraction of failed lookups is mostly constant when assuming a p_{gossip} above 60%. However, reducing gossiping probability p_{gossip} below 60% negatively affects the fraction of failed lookups as shown in Figure 22. Therefore, we propose a gossiping probability of 60% when using this first gossipping approach.

4.4.5 *Gossiping Mechanism Based on a Static Number of Data Sets*

Our second gossiping approach is based on a static number of data sets per *hello* message and a dynamic gossiping probability. We limit the number of data sets that may be attached to each *hello* message (s_{us}) in order to reduce the overall traffic. The transmitted data sets are selected randomly. This approach results in very predictable and homogeneous size of the *hello* messages and, therefore, the cross-layer traffic.

During this evaluation, we limit the number of data sets per *hello* message (s_{us}) to 5, 10, and 15. Once again, all other parameters are set to their default values (see Chapter 4.1.4 for further details).

4.4.6 *Results of the Gossiping Mechanisms Based on a Static Number of Data Sets*

The second gossiping mechanism is based on a static number of data sets per periodic *hello* message. In the following paragraphs, we discuss the results of the evaluation of this mechanism.

TRAFFIC

This gossiping mechanism limits the number of attached data sets per *hello* message directly. Therefore, this approach results in homogeneous cross-layer traffic in scenarios with two and four clusters as shown in Figure 24. Yet, the resulting overall traffic differs strongly and is virtually constant. Due to stale routing tables, messages have to be rerouted frequently in settings with a low s_{us}. Thus, even though we were

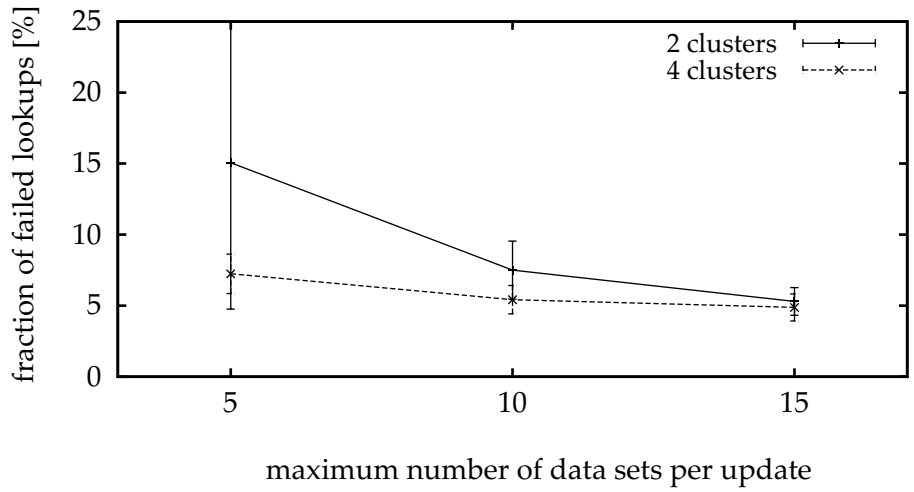


Figure 25: Evaluation results of the adapted *leaf set* gossiping mechanism

able to reduce the cross-layer traffic in these settings, new traffic is introduced at the overlay and, therefore, in the overall traffic.

FRACTION OF FAILED LOOKUPS

The scenarios with an increased number of clusters benefits from the small size of these clusters. As a result, the fraction of failed lookups is much lower in scenarios with four clusters compared to those using only two clusters as shown in Figure 25 especially when limiting the data sets to five sets per *hello* message. Furthermore, settings with two clusters and a small number of data sets per *hello* message strongly differ in the resulting fraction of failed lookups as indicated by the confidence intervals. As only few randomly selected data sets are transmitted per *hello* message, the reliability of the system strongly depends on the specific data sets that are transmitted.

For example, let us assume a setting with 50 nodes, two clusters, and a s_{us} of 5. Node A has four geographical neighbors and receives their *hello* messages every two seconds. In a worst case, all of those neighbors attach the very same five data sets to their *hello* messages and each of those geographical neighbors is included in the data sets of the other nodes. As a result, node A is only aware of six nodes in the *leaf set* (five nodes due to the data sets and the node itself). Due to the two clusters and 50 participants in the network, we assume that on average 25 nodes are located in the *leaf set*. Thus, node A is only aware of 24% of the nodes in the *leaf set* and, therefore, is in most cases not able to route a message correctly. However, in a best case scenario, we assume a node B also with four geographical neighbors. Each of these nodes sends a *hello* message that includes 5 nodes, which are neither a geographical neighbor of node B nor are included in the data sets of the other geographical neighbors. As a result, node B receives 16 different data sets from it's neighbors plus the identifiers of the geographical neighbors that are included in each *hello* message. As a result, node B is aware of 21 nodes (16 data sets, four virtual neighbors and node B itself) and therefore 84% of all *leaf set* nodes are periodically updated. However, as the data sets are selected randomly, the freshness of the routing table entries and, therefore, the fraction of failed lookups strongly differs between two different simulation runs.

	Basic setting	Optimized setting
Update interval of the data sets	2s	6s
Data set size	9 bytes	6 bytes

Table 5: Comparison of settings with an without the traffic reduction

4.4.7 *Reduced hello Message Size*

Each data entry in the *hello* message consists of an overlay identifier, an IP address, and a time to live field. As mentioned in Chapter 3.5.3, the suffix of the overlay identifier suffices to update the *leaf set*. Furthermore, as only a strongly limited amount of participants are assumed in a disaster relief scenario, neither leading bits of the IP address nor the identifier of the network are required to address a node. As a result, the traffic generated due to the update mechanism of the *leaf set* can be reduced by adapting the data sets. All other simulation parameters are set to their default values.

4.4.8 *Results of the Reduced hello Message Size Evaluation*

In the following paragraphs, we discuss the results of the simulation based evaluation of the *hello* message size reduction.

TRAFFIC

When reducing the size of the data sets as discussed in Chapter 3.5.3, the cross-layer traffic introduced due to the *leaf set* update mechanism can be reduced by approximately 35%. Even more important, the overall traffic introduced by the MP2P system is reduced by more than 22% when considering a scenario with 100 nodes and 4 clusters.

FRACTION OF FAILED LOOKUPS

The reduction of the *hello* message size does not affect the fraction of failed lookups. Due to this fact, reducing the size of the data sets is a very efficient way to reduce the overall traffic and can further be combined with any of the other mechanisms that have been discussed in the previous paragraphs.

4.4.9 *Summary of the Traffic Reduction Mechanisms*

All four mechanisms are able to reduce the traffic introduced by the *leaf set* update mechanism and, therefore, the overall traffic introduced by the Clustered Pastry system. Yet, reducing the size of the data sets, which are attached to each *hello* message, provides the best results when considering both, the resulting reduction in traffic and the fraction of failed lookups. Furthermore, this mechanism can be easily combined with any of the other three approaches.

Comparing the gossiping mechanisms with the adaptation of the update frequency reveals that adapting the update frequency provides more reliable results. Comparing the results of those three mechanisms shows that the best results can be achieved by increasing the update interval to 6s. These reduction mechanisms are, therefore,

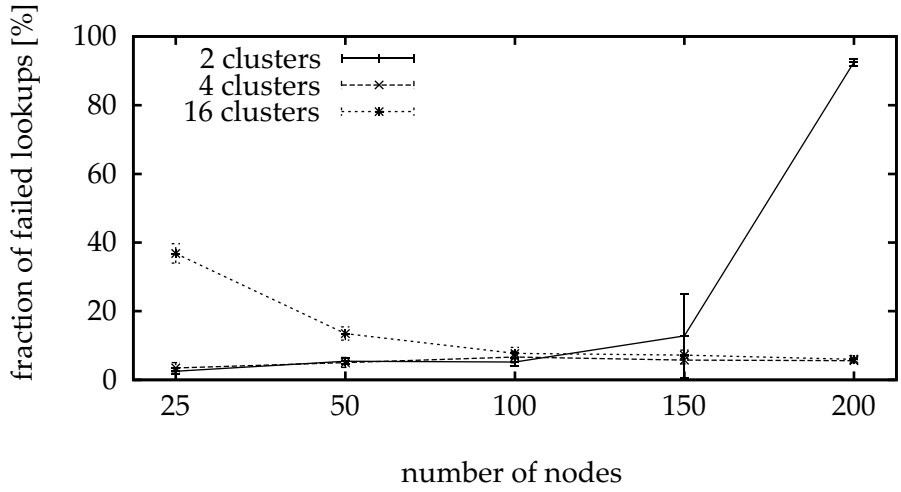


Figure 26: Fraction of failed lookups as a function of the number of participants

used in the following scenarios and are assumed as default parameters for the *leaf set* update mechanism.

4.5 SCALABILITY OF THE CLUSTERED PASTRY SYSTEM

As defined by our third evaluation goal, we have to analyze the Clustered Pastry system in the context of our disaster relief scenario. Therefore, the Clustered Pastry approach has to provide reliable services in scenarios with up to a few hundred nodes. In this section we analyze the scalability of Clustered Pastry and provide an evaluation of scenarios with up to 200 nodes. Furthermore, we vary the number of clusters in order to survey the impact of the cluster level on the efficiency of small and large scale scenarios.

In order to evaluate those scenarios two of the basic metrics are harnessed to analyze the influence of the number of participants on the reliability of the system: The fraction of failed lookups (f_{lookup}) and the overall traffic ($T_{overall}$) introduced by the Clustered Pastry system. Furthermore, the overall time (delay) required to complete a lookup is used as third metric to evaluate the usability and stability of the previously mentioned settings (t). In order to reduce the overall traffic, traffic reduction mechanisms as proposed in the previous section are harnessed. All other parameters are set to their default values.

4.5.1 *Results of the Scalability Analysis of Clustered Pastry*

Networks with an increased number of participants also introduce an increased amount of traffic. However, in the following paragraphs we discuss the evaluation results of scenarios with up to 200 nodes and analyze further the influence of the number of clusters on the traffic and fraction of failed lookups.

FRACTION OF FAILED LOOKUPS

In Figure 26 the fraction of failed lookup requests is shown as a function of the

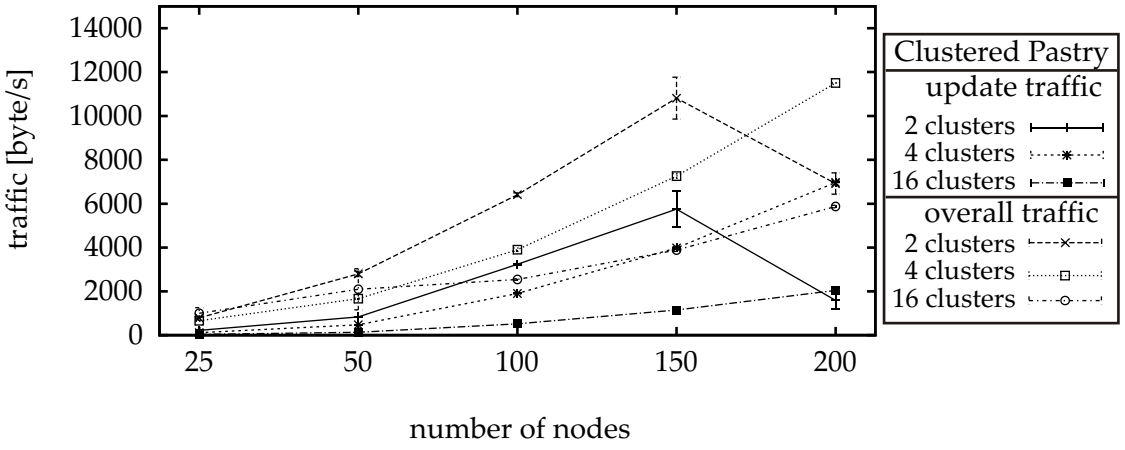


Figure 27: Cross layer and overall traffic of the Mobile Peer-to-Peer system as a function of the number of participants

network size and the cluster level. Most results of the clustered approach provide very good results with a loss rate of less than 5%. However, small scale scenarios with less than 100 nodes that are based on a setting with sixteen clusters perform poorly compared to systems with either 2 or 4 clusters. As a consequence of the increased number of clusters, nodes have to changes their ID more often. Due to this fact, also routing table entries become obsolete more often. This results in incorrectly routed lookup requests and, therefore, in an increased fraction of failed lookup operations. However, when the size of the network increases, the scenarios with a higher number of clusters provide better results. Due to the high number of nodes that are located in each cluster in large scale scenarios, a network with only two clusters cannot operate efficiently. On one hand, a high overhead is introduced by the *leaf set* update mechanism. On the other hand, the nodes of the *leaf set* are distributed over a large area. Therefore, update messages that provide information about nodes that have left or joined the network are delayed. This results in routing tables with incorrect entries. Even when we increase the update frequency to 2 seconds and, therefore, reduce the delay of the updates, no better results can be achieved as this increased update traffic results in congestion.

TRAFFIC

Due to the traffic reduction mechanisms introduced in the previous section, the traffic generated by the *leaf set* update mechanism has been strongly reduced. Yet, this update mechanisms still introduces a considerable fraction of the overall traffic of the Clustered Pastry system, especially when considering settings with a low clustering level and a high number of nodes. However, this overhead and, therefore, the overall traffic can be reduced by increasing the clustering level as shown in Figure 27. Yet, contrary to this expectation, the results of a scenario with 200 nodes and two clusters reveal a low traffic due to the update mechanism. As mentioned in the previous paragraph, large scale scenarios with a Clustered Pastry system that uses only a very limited number of clusters results in stale *leaf set* tables. Due to this fact, the data sets that are attached to the *hello* messages include only a subset of the nodes that are

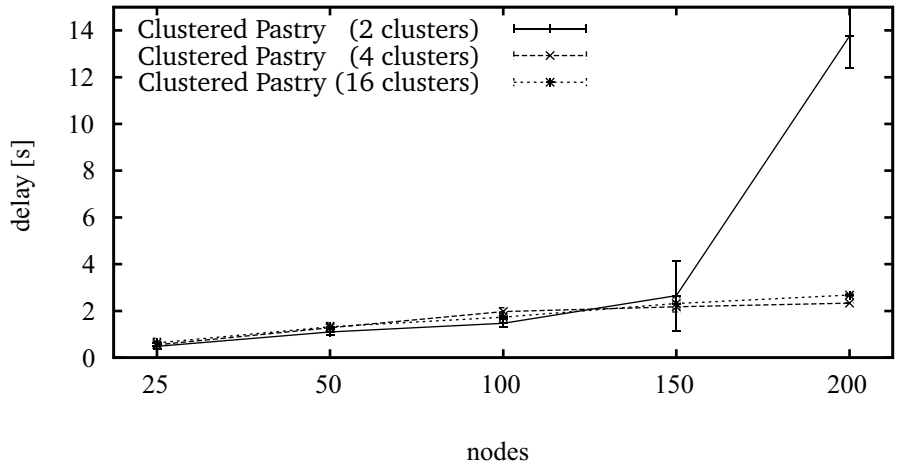


Figure 28: Overall delay of the lookup functionality as a function of the number of participants

located in the corresponding cluster. This results in a traffic reduction but also in a strongly increased fraction of failed lookups. On the other hand, small scale scenarios with a high cluster level introduce a relative high overall traffic. This is the result of the high number of cluster changes and the resulting traffic due to cluster based churn.

DELAY

The delay as shown in Figure 28 displays the average time required to successfully perform a lookup for an object. However, besides a metric for the usability of the system, the delay can also be harnessed to identify congested networks. Considering delay as a metric for the usability of a system is a straight forward approach as a system has to provide results in a reasonable amount of time. However, as most of our evaluation results provide a delay of less than three seconds, usability in terms of delay is not an issue for our Clustered Pastry system. Moreover, we use the delay as an indicator for congestion in the network. This is quite obvious when considering the scenario with 2 clusters and 200 nodes. In this scenario, the delay is strongly increased above 13 seconds due to the congestion introduced by the large clusters. However, in all other scenarios, we assume a low traffic load as the system is able to provide a reasonable delay for the lookup services.

4.5.2 *Summary of the Scalability Analysis of Clustered Pastry*

In this section, the scalability of the Clustered Pastry system has been evaluated. To sum the results up, our Clustered Pastry system is able to operate in small scale scenarios as well as in scenarios with 200 nodes as required by the first part of our third evaluation goal. Furthermore, we have shown that the clustering level can be used to adapt the system to the networks size and, therefore, to increase the efficiency of the system.

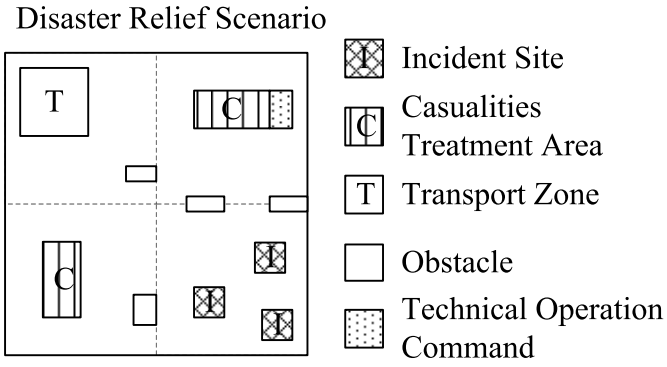


Figure 29: Preferred nodes in a scenario with proximity metric

4.6 INFLUENCE OF DISASTER RELIEF MOBILITY MODEL ON THE CLUSTERED PASTRY SYSTEM

The mobility of the participating nodes is one of the major challenges introduced by MP2P scenarios. The node mobility results in a dynamic topology and affects, therefore, the lifetime of routing table entries in the overlay and underlay. By now, only a basic mobility model based on a randomly generated mobility pattern has been used for the evaluation of our Clustered Pastry system. In this section a more complex mobility model is used to evaluate our Clustered Pastry system as required by our third evaluation goal.

Aschenbruck et al. [110] developed BonnMotion as a framework that provides a set of realistic mobility models including a model for disaster relief scenarios. This model split the deployment area in several areas, e.g., the incident site, the causalities treatment area, or the transport zone. A specific number of nodes are assigned to each of these areas. The mobility pattern of the nodes is derived according to these areas and observations made by Aschenbruck et al. at disaster sites. Yet, this mobility model is far more complex than the Random Waypoint model that has been previously used to derive the nodes mobility. Therefore, only medium sized scenarios with about 50 nodes can be simulated with a reasonable effort (simulation run-time below a month).

In order to evaluate our approach based on the disaster relief mobility model, we analyze the characteristics and outcomes of the simulations with the BonnMotion mobility model. Therefore, we analyze the number of cluster changes ($t_{cluster}$), the fraction of failed lookups (f_{lookup}) and the overall traffic ($T_{overall}$) generated by the Clustered Pastry system. The settings of the evaluation are based on the scenarios proposed by Aschenbruck et al. as shown in Figure 29.

4.6.1 *Results of the Influence of Mobility on the Mobile Peer-to-Peer System*

The BonnMotion disaster relief mobility model provides realistic mobility pattern of first responding units. In the following paragraphs we discuss the evaluation results of a setting that combines this mobility model with our Clustered Pastry system.

FREQUENCY OF CLUSTER CHANGES

The outcomes of the medium scale scenarios exhibit even slightly better results compared to scenarios based on the random waypoint model. Due to the mapping of

the nodes to specific disaster site areas, it is ensured that at least a fraction of these nodes do not leave the area they are assigned to. As those areas are mostly located in a single cluster, those nodes do not change their cluster. As a result, the number of cluster changes can be reduced by more than 50% when compared to a scenario that is based on the random waypoint model.

TRAFFIC AND FRACTION OF FAILED LOOKUPS

This also affects, on one hand, the overall traffic and, on the other hand, the fraction of failed lookups. Due to the reduced number of nodes, that change their cluster, traffic due to routing table updates is reduced. As a result, the overall traffic can be reduced by 8%. Both, the reduced number of cluster changes and the traffic reduction, further improve the probability to complete a lookup operation successfully. Therefore, the fraction of failed lookups is reduced to nearly 1%.

4.6.2 *Summary of the Influence of Disaster Relief Mobility Model on Clustered Pastry*

Considering a more realistic mobility model as provided by Aschenbruck et al., introduces participating nodes that are confined in their mobility to predefined areas. This results in a more stable network and reduces both, the fraction of failed lookups and the overall traffic generated by the MP2P system. Due to these facts, we assume that Clustered Pastry is able to handle the mobility patterns, which are introduced by disaster relief scenarios as required by our evaluation goal.

Moreover, the evaluated settings based on the BonnMotion disaster relief mobility model provides even better results as the random waypoint model. Thus, we assume that results evaluated based on a random waypoint model are at least comparable to results based on the disaster relief mobility model.

4.7 INFLUENCE OF THE TRAFFIC ON THE CLUSTERED PASTRY SYSTEM

In the previous scenarios, we used a fixed object size of 2kB. Even though this size would suffice to enable a storage of text based data and small pictures, we evaluate scenarios in this section with an increased object size with up to 15kB. Furthermore, all scenarios evaluated so far assumed that no other traffic is transmitted within the network. Yet, the MANET underlay may as well be used to transmit any other kind of data, e.g., for text messaging or voice communication. Therefore, we introduce background traffic in this section.

The background traffic is generated by using a basic UDP traffic generator as provided by the INET framework of OMNeT++. As a result, data packets with a size of 2kB are periodically sent by each node to a randomly selected node in the network. In order to increase the overall background traffic, the message interval is reduced. These scenarios are evaluated by using the basic metrics, the fraction of failed lookups (f_{lookup}) and the overall traffic ($T_{overall}$) generated by our Clustered Pastry system. All parameters of Clustered Pastry are set to their default values.

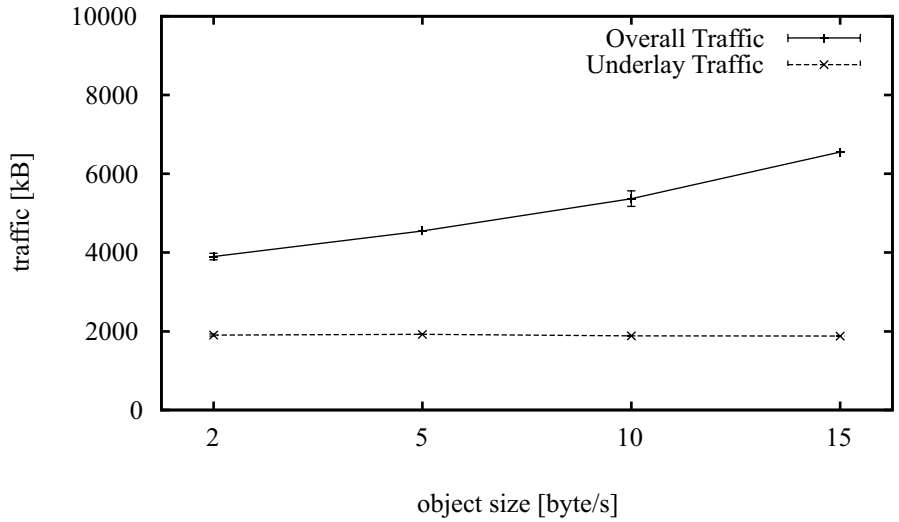
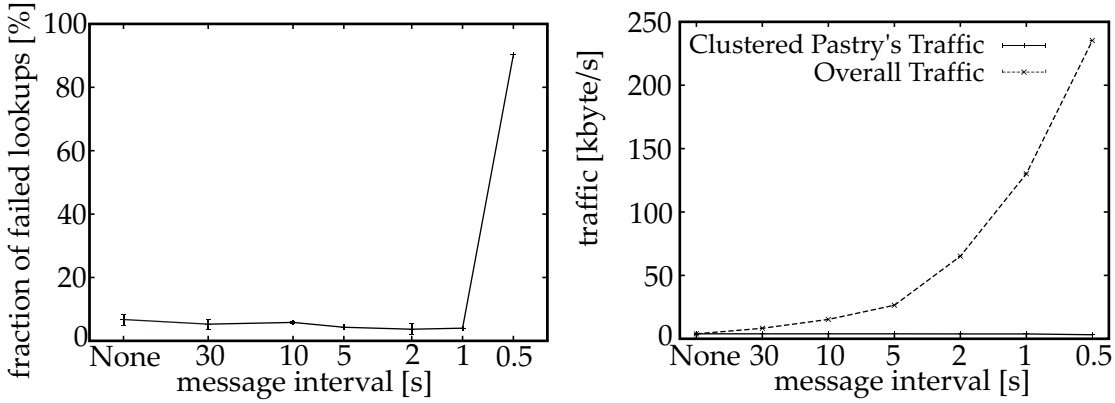


Figure 30: Influence of the object size on the traffic generated by Clustered Pastry



(a) Fraction of Failed Lookups

(b) Traffic

Figure 31: Impact of background traffic on Clustered Pastry

4.7.1 *Results of the Increased Data Object Size*

Increasing the object size obviously affects the traffic generated by the MP2P system. Though, the traffic increases by nearly 70% when increasing the objects size to 15kB as shown in Figure 30, the fraction of failed lookups is not affected. However, a size of 15kB should most probable suffice the requirements of our scenario as we assume that only text files and small sized pictures are stored with in the network in our scenarios.

4.7.2 *Results of the Effects of Background Traffic*

Background traffic like voice communication or text chatting may be introduced in disaster relief scenarios [57]. In the following paragraphs, we discuss the simulation results of settings with background traffic.

FRACTION OF FAILED LOOKUPS

Background traffic, introduced by other applications as voice transmission or text chatting affects the MP2P system due to the limited bandwidth. However, the fraction of failed lookup operations is only affected when a high background traffic is introduced as shown in Figure 31a. The reliability of the lookup mechanism of Clustered Pastry degrades only when the UDP messages are sent in an interval of less than two seconds.

TRAFFIC

Even when the background traffic is only sent twice a minute, the resulting traffic is almost equal to the overall traffic introduced by the Clustered Pastry system. When we consider that the Clustered Pastry is able to operate efficiently as long as the background interval equal or larger than two seconds, the overall traffic, which is transmitted via the MANET communication substrate is more than 16 times higher than the traffic of Clustered Pastry.

4.7.3 *Summary of the Influence of Traffic on Clustered Pastry*

As shown in this section, Clustered Pastry is also able to operate with objects of an increased size. Furthermore, a considerable amount of background traffic can be introduced without reducing the efficiency of the Clustered Pastry system. Due to this fact, we assume that other applications that generate a reasonable traffic can be used in parallel to our Clustered Pastry system as required by our last evaluation goal.

4.8 CHAPTER SUMMARY

In the previous chapter, we have introduced Clustered Pastry, a new cluster-based MP2P system that is based on a combination of a MANET and a DHT. As shown, this system is suited to provide reliable lookup functionality even when nodes are mobile and the topology is dynamic. In this chapter, we have evaluated and validated the efficiency of Clustered Pastry in a set of different scenarios.

We defined three major goals in the first part of this chapter that have to be satisfied in order to validate the efficiency of our Cluster Pastry system: The comparison of Clustered Pastry with a layered approach, the optimization of the algorithms of our Clustered Pastry system, and analyzing the impact of disaster relief specific settings on the Clustered Pastry system.

As a result of this evaluation, we were able to prove that our cross layered, clustered approach provides more reliable results than a basic layered approach that combines a DHT overlay with a MANET underlay. While the layered approach was not able to provide reasonable results in scenarios with more than 50 nodes, Clustered Pastry provides a reliable lookup services also in large scale scenarios with 100 nodes.

Furthermore, we were able to show the efficiency of a set of traffic reduction mechanisms that were introduced in the previous chapter. In the overlay, threshold areas can be used to reduce the effects of cluster related churn. As a result, the traffic generated by the overlay could be reduced by 10%. Beyond overlay traffic, we were able to reduce cross-layer traffic and, in particular, traffic introduced by the *leaf set*

update mchanism. By adapting the update frequency and the size of the update messages, we reduced the traffic generated by the *leaf set* updates to nearly 40%.

In this section we have further evaluated the efficiency of our approach considering specifications introduced by our disaster relief scenario. This includes the number of participating nodes, and background traffic introduced by other applications. As outcome of this evaluations we have shown that Clustered Pastry is able to operate in settings with 200 nodes. We were able to improve the efficiency of our system by adapting the number of clusters to the network scale. Moreover, settings with a specific disaster relief mobility model were evaluated. As a result, we have shown that our Clustered Pastry provides reliable results in settings with 50 nodes when using those specific mobility patterns. Furthermore, settings with background traffic were analyzed. As a result we were able to show that Clustered Pastry provides reliable services even when the overall traffic is strongly increased by other applications such as, e.g., voice communication or data transmissions.

Concludingly, we were able to show that our system provides reliable services even when considering background traffic or settings with up to 200 nodes. Clustered Pastry can easily adapt to the requirements of different networks by modifying the systems parameters, e.g., the cluster level when considering large or small scale scenarios or the threshold area to adapt to node mobility.

5

MALICIOUS BEHAVIOR IN MOBILE PEER-TO-PEER SYSTEMS

»Distrust and caution are the parents of security.«

— Benjamin Franklin

In the previous chapters, the Clustered Pastry system has been introduced and evaluated. This system provides reliable services despite the limited bandwidth and the highly dynamic topology of the MANET underlay. However, a benign behavior of all nodes that participate in the Clustered Pastry network was assumed by now.

Yet, nodes may get damaged due to the though environment of a disaster relief scenario. As a result, those nodes may fail to provide services that were requested by the network. In contrast to this unintentional malicious behavior, new challenges arise when nodes behave intentional maliciously. For example, malicious nodes may collude in order to achieve their goals. Those goals include selfish behavior. Selfish nodes benefit from services provided by the network, but are not willing to provide services to the network on their own. Furthermore, nodes may attempt to increase their gain at the expense of benign nodes or behave destructively in order to deny the network's services. In this thesis, we assume malicious, colluding nodes that behave destructively as a worst case scenario.

When introducing malicious nodes to the network, the availability of the services provided by the system cannot anymore be ensured. Therefore, vulnerabilities of MP2P systems are discussed in the first part of this chapter. In the second part, we focus on a subset of attacks, which have been identified as open challenges, and analyze the impact of these on our Clustered Pastry system [40].

5.1 SECURITY THREATS IN MOBILE-PEER-TO-PEER SYSTEMS

MP2P systems combine a P2P overlay with a MANET underlay. Therefore, attacks that were designed to affect those underlying networks can be used to attack an MP2P system as well. Plenty of attacks on those architectures have been identified in the last decade. In this section, the relevance of those attacks in the light of MP2P is briefly discussed. Furthermore, existing security mechanisms suggested by related work are considered in order to identify open challenges for the MP2P systems robustness.

5.1.1 *Malicious Behavior in the Underlay*

As mentioned previously, MP2P systems as our Clustered Pastry harness a MANET underlay. This architecture introduces several challenges concerning the networks security including a wireless channel, mobile nodes, and a decentralized structure.

MANETs are based on transmitting data via a wireless channel and, therefore, malicious nodes are able to eavesdrop on transmissions. Those passive attacks are hard to detect and information gathered by eavesdropping can be exploited to

prepare other attacks like spoofing attacks [117]. Furthermore, wireless transmissions provide only limited resources in terms of bandwidth [31]. This limitation has to be considered when designing security mechanisms. Moreover, due to the nodes' mobility, permanently changing geographical neighbors have to be assumed [31]. This introduces challenges in the area of trust management [68]. MANETs also have to rely on intermediate nodes in order to communicate with nodes that are not within the transmission range. These characteristics of the decentralized, mobile system introduce several challenges for MANET and, therefore, MP2P security.

Several attacks on MANETs have been discussed in the related work. Yet, also plenty of security mechanisms have been developed to prevent or reduce the effects of those attacks on the network in the last decade. However, mechanisms have been proposed to ensure that the security goals are adhered to by, e.g., using cryptography [53] [93] [119], or harnessing an IDS or an Intrusion Response System (IRS) [63] [72].

Most MANETs and, therefore, most security mechanisms proposed for this architecture were not developed for a specific application. As a result, we assume that most of the security mechanisms developed for MANETs can also be used to ensure the robustness of the MP2P systems underlay. Therefore, we assume that the underlay of our MP2P system is secured by state of the art mechanisms and, therefore, provides reliable services.

5.1.2 *Malicious Behavior in the Overlay*

DHTs as Pastry are vulnerable against a wide range of attacks due to their decentralized and cooperative characteristics. In the following paragraphs, we briefly discuss those security challenges that also have to be considered in the context of MP2P system.

The decentralized structure of P2P systems introduces challenges such as providing a reliable access control or authentication mechanism [67], especially when considering scenarios where no central entity is available. Furthermore, malicious behavior such as Sybil attacks [24] are hard to detect in a decentralized overlay. By performing a Sybil attack, a malicious nodes bootstraps multiple times in the same network. Therefore, this single Sybil node is able to control multiple virtual overlay nodes. As a result, a malicious node is able to strongly increase the impact on the network when performing an attack. Furthermore, DHTs have to rely on other nodes that are located within the network to resolve lookups, to store and retrieve objects, and to update routing tables. During routing, intermediate nodes are required to forward the request to the destination. Due to malicious behavior or as a result of a node failure, those request messages may be dropped or redirected. Furthermore, root nodes are required to store objects and are contacted whenever this object is requested. A malicious behavior of those root nodes may result in a denial of each locally object stored.

However, multiple security mechanisms were developed for P2P systems in the last decade. This includes decentralized access control and authentication mechanisms. Those mechanisms are based, e.g., on threshold cryptography [97]. Threshold cryptography can be used to generate signatures in a decentralized way and has been adapted for admission control in P2P networks [64]. Furthermore, mechanisms were introduced by the related work to increase the robustness against Sybil attacks. Those

mechanisms proposed for DHTs are mostly based on introducing high costs for the generation of an overlay identity, e.g., by using crypto puzzles [17]. As a result, the number of Sybil identities a node is able to generate were limited by the available resources of the malicious node. Moreover, security mechanisms introduced for P2P systems that considers the security and robustness of the routing mechanisms are mostly based on introducing redundancy.

Even though most security challenges have been widely discussed and multiple security mechanisms have been proposed in the recent years, limitations introduced by the MANET underlay to the overlay were not considered. Therefore, some of these mechanisms cannot be used efficiently in MP2P scenarios.

5.1.3 *Related Work and Open Challenges in the Field of Mobile Peer-to-Peer Security*

In Chapter 2.2.1, we discussed the six basic security goals as defined by the related work. These goals include the *confidentiality*, the *integrity*, the *availability*, the *authentication*, the *non repudiation*, and the *privacy* of a system and the services provided by this system, respectively. We harness those security goals to identify open challenges in MP2P security. Therefore, we discuss the security mechanisms that have been developed for MP2P systems (see also Chapter 2.2.6) and mechanisms introduced in the previous paragraphs in the light of those security goals in the following paragraphs.

These approaches focuses on access control, replication distribution mechanisms, or privacy issues and are summarized as follows. Both, Čapkun et al. [112] as well as Fenkam et al. [26] introduce access control mechanisms that can be used in the light of MP2P scenarios. These mechanisms are required to ensure that only authorized nodes access data provided by the network (*confidentiality*). Privacy issues in MP2P networks have been studied by Maniulus [71] as well as Han and Liu [43]. Even though *privacy* is not required in order to ensure the reliable functionality of the MP2P system, it may be important in disaster relief scenarios as, e.g., medical informations of causalities may be stored in the network. Furthermore, a basic replication mechanism has been introduced by Mondal et al. [75] [76] in the context of MP2P systems. Yet, this approach is based on *super-peers*. As Clustered Pastry does not provide *super-peers* due to the structure of the overlay and due to the requirements of the scenario, this replication mechanism cannot be used. However, a replication mechanism is required to ensure the *availability* of the services provided by the overlay [62] [86].

Certain types of attacks can be performed on both, the underlay as well as the overlay. For example, the Sybil attack affect MANETs and P2P networks. Multiple security mechanisms have been proposed for both architectures. However, it suffices to identify such an attack on one of the underlying architectures. Therefore, the most efficient security mechanism proposed for either MANET or P2P networks is harnessed to detect and/or prevent those types of attacks. In the case of Sybil attacks, promising approaches have been introduced for MANETs that are based analyzing overheard messages [78] [81]. Due to this fact, we assume that Sybil attacks can be identified by existing underlay mechanisms and, therefore, are neglected in the context of this thesis.

By now, basic mechanisms to ensure *confidentiality* and *privacy* have been introduced in the related work. Furthermore, cryptographic mechanisms can be used to *authenticate* peers. Those cryptographic mechanisms can also be used to sign stored objects

and update messages. As a result, the *non repudiation* as well as the *integrity* of the MP2P system can be ensured. Therefore, a decentralized cryptographic mechanism as the previously mentioned threshold cryptography can be used to generate asymmetric keys for newly booted nodes. Another promising approach is based on identity based encryption [98]. By harnessing this approach private keys may be generated by a central entity based on a given public key. As a result, the static suffix of the nodes overlay identifier may be used as public key and the private key may be generated by this central instance in prior to the disaster relief operation. However, a large set of cryptographic algorithms are readily available and, therefore, not the focus of this thesis.

All of those mechanisms (except the mentioned replication mechanism) neglected the *availability* of the services of the network. Yet, this security goal is essential when considering a disaster relief scenario as mentioned earlier. Therefore, attacks on the availability of the networks services is discussed in the rest of this chapter. This includes a malicious behavior of the root of an object and routing attacks on the overlay. Both of these attacks have an high impact on the availability of the service provided by the system.

5.1.4 *Summary of the Open Challenges in Mobile Peer-to-Peer Security*

MP2P systems are vulnerable against multiple attacks as discussed in this section. However, on one hand, some security mechanisms have already been developed for MP2P systems that cover a part of the discussed security challenges. On the other hand, some mechanisms can be inherited from the underlying architectures, in particular mechanisms that have been developed for MANET systems.

However, attacks on the overlays routing mechanism have been identified as a major challenge. Those attacks have been neglected in the context of MP2P systems. Furthermore, security mechanisms developed for static DHTs are based on introducing redundancy [6] [17] [51] [58] [101] or harness assumptions that do not apply to MP2P scenarios [17] [30] [114]. Due to limited bandwidth, redundancy introduced by the security mechanism must be minimized. Apart from routing attacks, maliciously behaving root nodes are also discussed in this thesis. The impact of the resulting attacks can only be minimized by using replicas. Even though plenty of replication schemes for overlay and underlay systems have been developed recently, we assume that an replication mechanism, which has been adapted to the characteristics of the MP2P scenario, is more efficient. Yet, existing adapted approaches are based on superpeers. As no superpeers are provided by DHTs, novel replication mechanisms are required.

5.2 AFFECTING THE AVAILABILITY OF MOBILE PEER-TO-PEER SERVICES

Each node in a DHT is a supplier and consumer of services. MP2P systems, as considered in this thesis, provide storage and retrieval services in a decentralized way. Until now, benign behavior of each node that participates in the network is assumed. Therefore, a lookup may only fail as a result of collisions, due to fading effects in the wireless underlay, or when the requested object is not available at the root node.

However, in a realistic scenario, malicious behavior has to be considered. As discussed in the previous section, multiple mechanisms have already been proposed for MP2P by the related work to ensure the *confidentiality*, *privacy*, *integrity*, and *non repudiation* of MP2P systems. Yet, the *availability* of services has been mostly neglected by now. Therefore, attacks on the *availability* of objects are highly relevant for this thesis. In the rest of this section, these attacks on the services of MP2P systems will be discussed in detail. Moreover, we provide analytic models to determine the impact of those attacks on the network.

5.2.1 *Storage and Retrieval Attack*

Whenever a node wants to store an object in the MP2P network, the root node of this object is determined via a lookup operation. Thereafter, either a link to the node that provides this object or the object itself is locally stored at the root node. When this object is requested by any other node in the network, the root node is contacted and the object is retrieved thereafter.

However, a benignly behaving root node is assumed that is willing to store and to supply objects on demand. Yet, malicious root nodes may not provide these services or, at least, may not provide these services to every node in the network. A maliciously behaving root node may either deny stored objects or reply a faulty object when a request is received. As a result, those *Storage and Retrieval Attacks* [99] are able to strongly affect the reliability of an MP2P system.

ANALYTIC EVALUATION

The impact of this attack obviously depends directly on the fraction of malicious nodes that participate in the network (f). Therefore, when losses due to the networks characteristics are neglected, the probability of a successful lookup (σ_{SARA}) can be derived as shown in Equation 5.1.

$$\sigma_{SARA} = 1 - f \tag{5.1}$$

Even though no data object can be retrieved due to the malicious behavior of the root, a complete overlay routing has to be performed. Due to this fact, the introduced traffic in terms of sent messages (m_{SARA}) and the required number of overlay hops (h_{SARA}) is similar to a successful lookup (h) (as shown in Equation 5.2). When the malicious node does not deny the object but transmit a fault copy, transmission related traffic is also generated by this attack.

$$h = h_{SARA} = m_{SARA} \tag{5.2}$$

STORAGE AND RETRIEVAL ATTACKS IN MOBILE PEER-TO-PEER SYSTEMS

The *Storage and Retrieval Attack* [99] is a simple and efficient attack. The services provided by the MP2P system are strongly affected when requested objects are denied by maliciously behaving root nodes. Furthermore, the source of such a request is not able to detect whether the root node behaves maliciously, the object is unavailable in the network or a faulty object has been stored.

Yet, a high amount of nodes have to perform the *Storage and Retrieval Attack* in order to introduce a strongly increased number of failed lookups. However, when combined with other attacks as the Sybil attack, the impact of this attack can be strongly increased. Furthermore, when nodes are able to select their node identifier and, therefore, the identifier space they are responsible for, *Storage and Retrieval Attacks* can be used to deny specific objects. Moreover, as only a limited number of nodes are assumed in MP2P scenarios, the probability that a malicious node that tries to deny a specific object is responsible for this object is quite high compared to a static scenario based on the Internet as underlay with more than 10,000 nodes.

5.2.2 *Incorrect Lookup Routing Attack*

Nodes in a DHT have to rely on the benign behavior of other participants in the P2P network during a lookup. The *Incorrect Lookup Routing Attack*, which has been introduced by Sit and Morris [99], exploits this requirement to deny the services provided by the DHT. As a result of this attack, route requests are dropped or redirected by malicious intermediate overlay nodes. While dropping requests is a straightforward approach, redirecting messages to other malicious nodes is less conspicuous and harder to detect by countermeasures. Nevertheless, most DHTs are unable to detect either kind of this attack and, therefore, we focus on the packet dropping variant of the *Incorrect Lookup Routing Attack* in this chapter. This attack is most probable more efficient in terms of the number of failed lookups and, moreover, less complex as no colluding nodes are required. In the following paragraphs, the impact of this attack is evaluated by theoretical means and discussed in both, a P2P as well as an MP2P context.

ANALYTIC EVALUATION

Most DHT routing algorithms are based on a recursive approach. Requests are, in each overlay hop, forwarded by intermediate nodes to a node closer to the destination regarding the virtual distance in the overlay identifier space until the destination is reached. This algorithm is very efficient regarding the overhead generated due to a lookup. Yet, this approach does not result in a robust lookup mechanism when under attack. Castro et al. [17] derived an equation to estimate the probability of a successful lookup process σ_{ILR}. This equation is based on the average number of overlay hops h and the fraction of malicious nodes f as shown in Equation 5.3.

$$\sigma_{ILR} = (1 - f)^{h-1} \tag{5.3}$$

The number of the average overlay hops that are required to complete a lookup is defined by the specific P2P routing algorithm. As mentioned in Chapter 2.1.3 Pastry introduces an average hop count (h_{Pastry}) that is a logarithmic function of the network size (N) and the parameter b as shown in Equation 5.4.

$$h_{Pastry} = \log_{(2^b)}(N) \tag{5.4}$$

Our Clustered Pastry MP2P approach on the other hand has a hop count (h_{CP}) that based on the number of overall clusters and on the parameter b as shown in Equation 5.5.

$$h_{CP} = 1 + \log_{(2^b)}(C) * (1 - \frac{1}{2^b}) \tag{5.5}$$

Besides the fraction of lost requests, the number of sent messages (m_{ILR}) per request and, therefore, the number of required hops must be considered. Those two metrics correlate and can also be used to estimate the introduced traffic per request. Furthermore, we have to differentiate between the number of average hops of a successful request (h) and the number of an average hops (h_{ILR}) that also includes failed requests. In a recursive scenario without any security mechanisms, the average number of sent lookup messages (m_{ILR}) is equal to the number of hops that includes the failed requests h_{ILR} as shown in Equation 5.6.

$$h_{ILR} = m_{ILR} = \sum_{i=0}^{h-1} (1 - f)^{i-1} \tag{5.6}$$

DISCUSSION

Due to the misbehavior of intermediate nodes, the availability of objects and services, provided by the MP2P system can be strongly affected. Only a limited number of malicious nodes suffice to affect the reliability of the recursive routing algorithm strongly, as discussed in the previous paragraph. Furthermore, the source of a lookup is unable to detect the maliciously behaving node due to the recursive structure of the lookup algorithm. Hence, a second request message may be dropped by the very same malicious node.

The effect of this attack can further be increased when combined with other attacks that increase the probability that a lookup message is routed via a malicious node. Therefore, Sybil attacks as well as a manipulation of *routing tables* can be performed in prior of the attack to increase the overall impact of the *Incorrect Lookup Routing Attack*.

5.2.3 *Forging Reply Messages*

In order to retrieve an object from a DHT, the source of this lookup has to hash the requested objects name to obtain the objects overlay identifier. Thereafter, a lookup for this overlay identifier has to be initiated in order to determine the root of this object. In most cases intermediate overlay nodes are required to forward this request to the root node. Therefore, the *routing tables* are used to determine a link to the node that is logically closest to the overlay identifier in order to forward the request towards its destination until the root node receives this request.

However, this mechanism introduces a major drawback. The source of the lookup is unaware of the root node's overlay identifier but only is aware of the object's overlay identifier. By forwarding the request to the virtually closest known node, each intermediate node may claim to be the root of the object and submit a reply message. Instead of replying the requested object, this malicious intermediate node may either provide a faulty object or claim that this object is not available in the network.

ANALYTIC EVALUATION

This attack affects the network in a very similar way as the *Incorrect Lookup Routing Attack*. Both attacks are initiated by malicious intermediate nodes. Furthermore, a single intermediate node suffices to result in a failed lookup. Due to this fact, the same analytic models can be used to determine the impact of this attack (Equation 5.3 and 5.6).

Yet, those attacks differ in the way a maliciously behaving node reacts on an incoming request message. The *Incorrect Lookup Routing Attack* results in failed lookup due to the dropped request message. By forging a reply message, the source node assumes that the lookup has not failed, but the requested object is faulty or not available in the network.

DISCUSSION

Forging reply messages results in a denial of the lookup functionality. Yet, as the request message is not simply dropped by the malicious node, but a reply message is sent, it is hard to differ for the source of the request whether a lookup has been successfully completed but the object is unavailable or if the reply message has been forged.

This behavior may as well be a result of faulty or stale routing tables. Due to this, not always a malicious behavior has to be assumed. Furthermore, the impact of this attack can be increased in a similar way as introduced for the *Incorrect Lookup Routing Attack*.

5.2.4 *Summary of Routing Attacks and Maliciously Behaving Root Nodes*

Three attacks have been introduced in this section that are able to strongly affect the availability of the networks services. The first attack is based on denying requested objects. As a result, this Storage and Retrieval Attack does not respond to received lookup request or replies a fault object. Thus, only root nodes are able to perform this attack. Both, the second and the third attack, which have been discussed in this section, are based on maliciously behaving intermediate nodes. Due to the structure of the routing tables of a DHT, intermediate nodes are required whenever a lookup is initiated. Those intermediate nodes have to forward the lookup towards the destination. Yet, maliciously behaving nodes exploit this requirement to deny the networks services. Thus, whenever a lookup request is received by a maliciously behaving node this message is either dropped (Incorrect Lookup Routing Attack) or a forged reply message is sent (Forging Reply Messages). However, those two strongly differs on the outcome of this attack. The *Incorrect Lookup Routing Attack* results in a failed lookup as no reply message is received by the source node. Yet, when forging a reply message, the source node assumes that the lookup has been successful but the object is not available in the network.

5.3 EVALUATION

In this section, the previously discussed attacks are analyzed and evaluated in the context of our Clustered Pastry system. After analyzing the impact of each of the previously discussed attacks, the effects of a combination of these attacks is evaluated.

The evaluation of the maliciously behaving nodes is based on the already introduced analytic models, on simulation and, in the case of a combined attack, on testbed results.

5.3.1 *Evaluation Settings and Metrics*

In the following paragraphs, the evaluation goals are defined and the evaluation metrics are introduced. Furthermore, evaluation settings as well as used evaluation methods are discussed.

GOAL OF THE EVALUATION

Our first goal of this evaluation is to determine the impact of the previously discussed attacks on our Clustered Pastry system. Therefore, we simulate settings with maliciously behaving intermediate nodes, root nodes and a combination of both.

The analytical models provide a initial indication of how maliciously behaving nodes may affect the availability of the network's services. Yet, those models were developed for static wireless scenarios and do not consider packet loss due to the wireless characteristics of the MANET underlay. Therefore, our second goal is to validate these models in the context of MP2P scenarios.

Moreover, the update mechanism of Pastry's *routing table* may introduce a vulnerability, which may be exploited by maliciously behaving nodes. Clustered Pastry may inherit this weakness as the overlay is based on the Pastry DHT. Therefore, this vulnerability should be revealed as our third evaluation goal in this chapter.

EVALUATION METRICS

The *Storage and Retrieval Attack*, the *Incorrect Lookup Routing Attack* and the *Forging of Reply Messages* share a common goal. All of these attacks affect the availability of the storage and retrieval services provided by the MP2P system. Those maliciously behaving nodes either exploit the vulnerabilities of the lookup mechanism or deny stored objects. Due to this fact, we harness the fraction of failed lookups (f_{lookup}) as a basic metric to determine the availability of stored objects in the network.

Moreover, the maliciously behaving nodes that perform the *Forging of Reply Messages* or the *Incorrect Lookup Routing Attack* affect the routing algorithm of Clustered Pastry. Therefore, we analyze the traffic (T) generated by the MP2P system, in particular the data amount transmitted during simulation and the average number of underlay hops per lookup (T). Those metrics are used to identify the impact of the attacks on the lookup mechanism.

PARAMETERS AND SETTINGS OF THE SIMULATION BASED EVALUATION

In order to evaluate the impact of maliciously behaving nodes on the Clustered Pastry system, the OMNeT++ [111] simulator as introduced in Chapter 4.1.3 is used. Moreover, we implemented the three previously discussed attacks. The maliciously behaving nodes are selected randomly and perform the selected attack only. For example, a malicious node that performs the *Storage and Retrieval Attack* behave benignly during routing and when a object has to be stored locally, but behaves malicious when any stored object is requested by any node in the network.

If not stating otherwise, we use default parameters for the Clustered Pastry system and the simulated environment as defined in Chapter 4.1.4. Furthermore, traffic reduction mechanisms are used as proposed in Chapter 3.5.3. We further varied the fraction of malicious nodes from 0% up to 50% during the evaluation of the routing attacks.

PARAMETERS AND SETTINGS OF THE TESTBED-BASED EVALUATION

The combined attack has also been evaluated by two wired Testbeds in order to determine the characteristics of the overlay in a realistic scenario free from the influences of the wireless channel. Therefore, the FreePastry [88] implementation of the Pastry P2P system is used as Clustered Pastry is based on this DHT and, therefore, those two systems maintain similar routing tables, and use the same routing algorithm. Default settings of this FreePastry (parameter $b = 4$, *leaf set* size of 32 nodes) has been used to evaluate the combined attack. To provide storage and maintenance services to the network, FreePastry was extended by the PAST [89] application.

PlanetLab[1] is a worldwide testbed for development and deployment of prototypes in a real-world environment. Larry Peterson (Princeton) and David Culler (UC Berkeley and Intel Research) initiated this project in 2002. Today, PlanetLab consists of more than 1150 nodes distributed around the world on nearly 550 sites. While the hardware and the load of every node differ widely, the PlanetLab software used to manage access to the testbed and the operating system (Linux Fedora based) is homogeneous.

The G-Lab[2] project started in 2008 and ended in 2012. The major objective of this project, which was funded by the German Federal Ministry of Education and Research (BMBF), was to create a national network to develop and evaluate future Internet technologies. More than 150 nodes distributed over Germany at multiple universities (including Berlin, Darmstadt, Karlsruhe, Munich, Kaiserslautern, Würzburg) were online. These nodes were equipped with homogeneous hardware and the overall load was low. For management, the PlanetLab software was deployed such that experiments designed for PlanetLab could be conducted on G-Lab without adaptations.

5.3.2 *Evaluation of the Storage and Retrieval Attack*

Maliciously behaving nodes are able to deny the access to each object that is stored locally. In the following paragraphs, we analyze the impact of the *Storage and Retrieval Attack* according to our first evaluation goal. Moreover, evaluation results are compared to the previously discussed analytic model to meet the requirements of our second goal.

The *Storage and Retrieval Attack* can only be performed by root nodes. Therefore, we assume a benign behavior of the source and each intermediate node in following evaluation. Maliciously behaving root nodes reply faulty objects whenever they receive a lookup request. Those faulty objects have the same size as the original objects but differ in the provided data.

1 http://www.planet-lab.org/
2 http://www.german-lab.de/

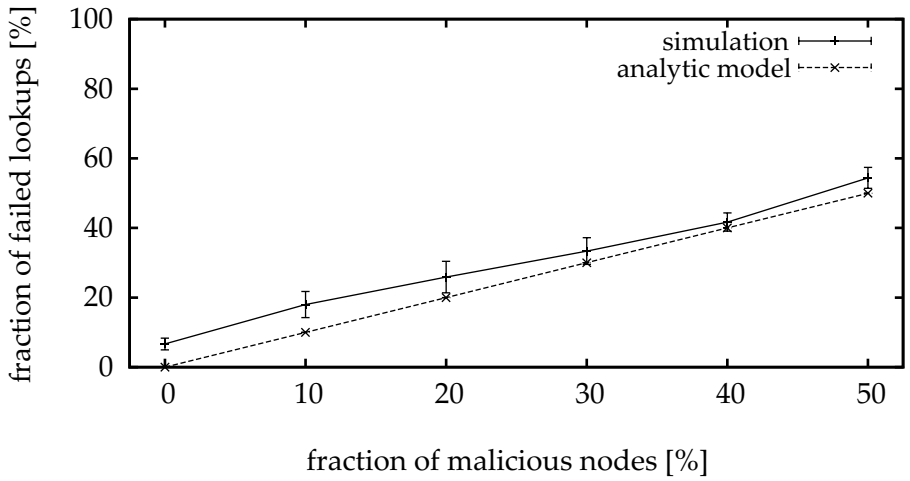


Figure 32: Influence of the *Storage and Retrieval* attack on the fraction of failed lookups

In order to evaluate the *Storage and Retrieval Attack*, three metrics are used: The fraction of failed lookups (f_{lookup}), the overall traffic generated by the Clustered Pastry system ($T_{overall}$) and the average number of underlay hops (T_{hops}).

RESULTS OF THE STORAGE AND RETRIEVAL ATTACK EVALUATION

As long as the object and node identifiers are equally distributed in the overlay identifier-space, the average fraction of failed lookups (f_{lookup}) due to this attack converges to the overall fraction of malicious nodes in the network as depicted in Equation 5.1. However, the simulation results introduces a slightly increased loss due to the characteristics of the wireless MP2P network as shown in Figure 32. Apart from that, simulation based results validate the analytic model discussed previously.

The traffic (T) introduced to the network by the *Storage and Retrieval Attack* strongly depends on the specific implementation of this attack. When malicious nodes drop request messages or reply that the requested object is not available, the overall traffic ($T_{overall}$) is reduced as no object is transmitted. Yet, when the root node replies a faulty object the overall traffic is comparable to a benign scenario, even though the requested object has not been received. Due to this fact, the overall traffic is not affected by the *Storage and Retrieval Attack* as we assume that maliciously behaving root nodes reply fault objects in our scenarios. Moreover, the average number of overlay and underlay hops (T_{hops}) per lookup is also not affected as the routing process itself is not influenced by this attack as predicted by the analytic model.

SUMMARY OF THE STORAGE AND RETRIEVAL ATTACK

We have evaluated the impact of *Storage and Retrieval Attack* on our Clustered Pastry system as demanded by our first evaluation goal. Moreover, we were able to validate the analytic models of this attack, which have been introduced previously, according to our second goal.

However, as long as no replicas are distributed in the network, all objects that are stored at a maliciously behaving root node are unavailable for each participant in the Clustered Pastry network. Thus, the *Storage and Retrieval Attack* is a very challenging attack, even though a high fraction of malicious nodes is required in order to deny

most of the lookup requests initiated in the Clustered Pastry system. The impact of the *Storage and Retrieval Attack* can further be strongly increased when performing this attack in combination with, e.g., the *Incorrect Lookup Routing Attack*. Therefore, it is highly recommended to use replication mechanisms in order to introduce robustness against attacks of maliciously behaving root nodes.

5.3.3 *Evaluation of the Impact of Maliciously Behaving Intermediate Nodes*

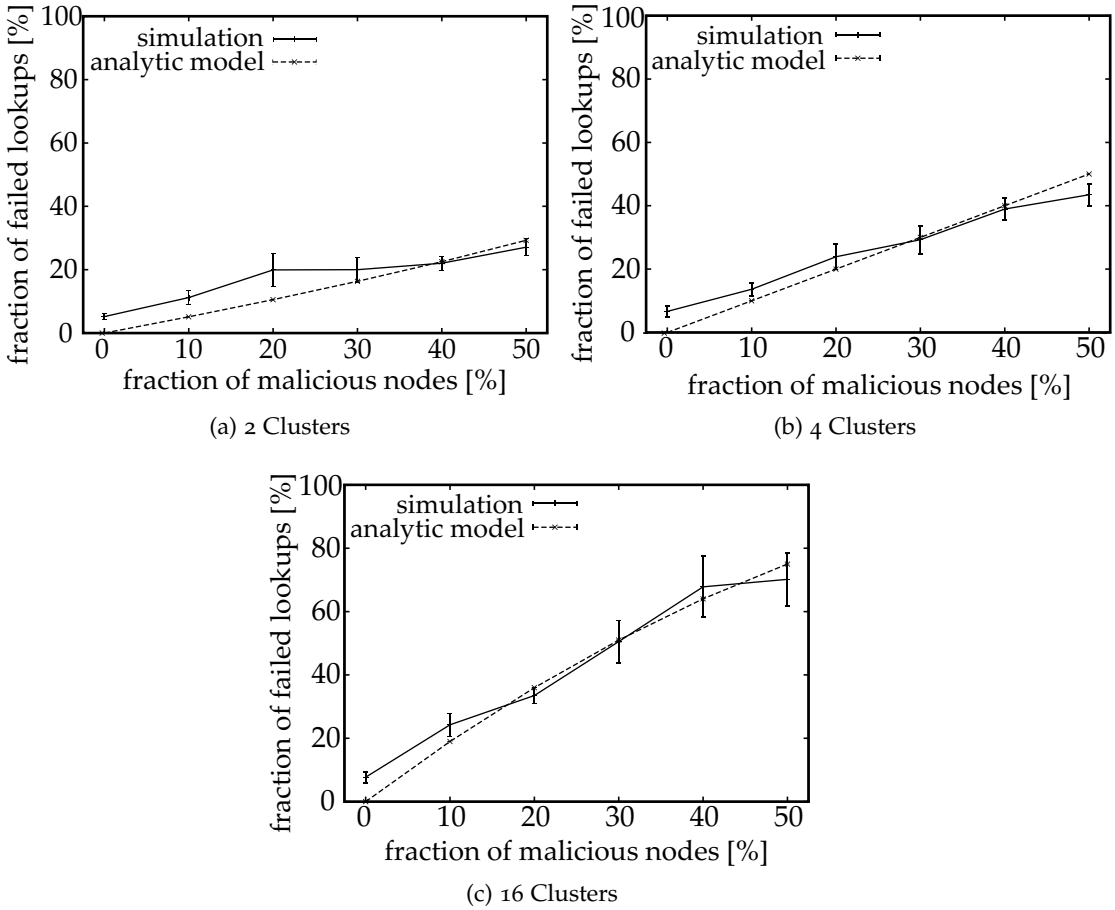
Both, the *Incorrect Lookup Routing Attack* as well as *Forging Reply Messages* are performed by intermediate nodes during the routing process. Sender and destination of the request message are assumed as benignly behaving nodes. As a result of both attacks, the routing request fails. Hence, both attacks affect the network in a similar way even though these attacks differ regarding the outcome of the lookup. However, due to these similarities, both attacks are evaluated together in the following paragraphs in order to satisfy the requirements of our first evaluation goal. According to our second goal, we will further harness the results of this evaluation to validate the analytic models of those attacks.

We assume that the impact of those attacks strongly depend on the average number of overlay hops as mentioned earlier. As the number of overlay hops is a function of the number of clusters, we vary this parameter during the following evaluation. Furthermore, we use the fraction of failed lookups (f_{lookup}) and the average number of underlay hops (T_{hops}) as metric to determine the impact of those attacks on our Clustered Pastry system.

RESULTS OF THE EVALUATION OF MALICIOUSLY BEHAVING INTERMEDIATE NODES

The results of the evaluation of maliciously behaving intermediate nodes confirm our assumption regarding the dependency of the impact of these attacks and the number of clusters in the network (see also Chapter 5.2.2). As shown in Figure 33a, the fraction of failed lookups (f_{lookup}) in scenarios with two clusters is low compared to the outcomes of scenarios with an increased number of clusters as shown in Figure 33b and Figure 33c. Especially scenarios with 16 clusters are strongly affected. For example, approximately 40% of all lookups fail in a scenario where 20% of the participating nodes behave maliciously. The evaluation results only slightly differ from the predicted fraction of failed lookups as the models neglect the characteristics of the lossy wireless channel. However, the number of clusters cannot be chosen freely but depends on the number of participants as shown in Chapter 4.5. As a result, networks with an increased number of participants are highly vulnerable to maliciously behaving intermediate nodes.

As request messages are intercepted by maliciously behaving intermediate nodes in this scenario, the overall number of overlay and underlay hops (T_{hops}) is affected by the fraction of malicious nodes. The overlay hop count behaves as predicted by Equation 5.6. Yet, the number of underlay hops is only slightly decreased even in scenarios with a high fraction of malicious nodes as shown in Figure 34. As mentioned in Chapter 3.5, a lookup requested is routed with each hop geographically closer to the root node of the requested object. As a result, the number of underlay hops that is required per overlay hop is reduced with each subsequent routing step. Due to this



(a) 2 Clusters

(b) 4 Clusters

(c) 16 Clusters

Figure 33: Impact of the *Incorrect Lookup Routing Attack* on the Clustered Pastry system as a function of the number of clusters

fact, the number of underlay hops is, therefore, only slightly decreased as a result of the attack.

SUMMARY OF THE MALICIOUSLY BEHAVING INTERMEDIATE NODES

We discussed the evaluation of maliciously behaving intermediate nodes as required by our first evaluation goal. Attacks initiated by intermediate nodes as the *Incorrect Lookup Routing Attack* and the *Forging Reply Messages* have a high impact on scenarios with many participants and, therefore, a large number of clusters. Furthermore, the overall traffic of a network under attack has not been considerably reduced due to the structure of the Clustered Pastry system. The outcomes of the evaluation moreover validates the analytical models as required by our second evaluation goal. However, both of these attacks can further be combined with the *Storage and Retrieval Attack* as will be discussed in the following section.

5.3.4 *Evaluation of the Combined Attack*

By combining two of the previously introduced attacks we assume that we are able to increase the overall impact of the maliciously behaving nodes. Therefore, we evaluate a scenario where malicious nodes perform both, the *Incorrect Lookup Routing*

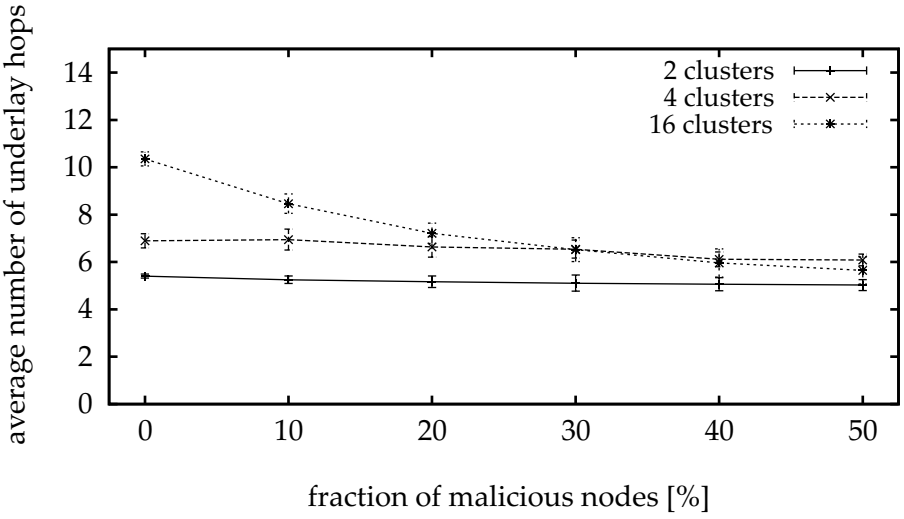


Figure 34: Influence of the *Incorrect Lookup Routing Attack* on the average number of underlay hops

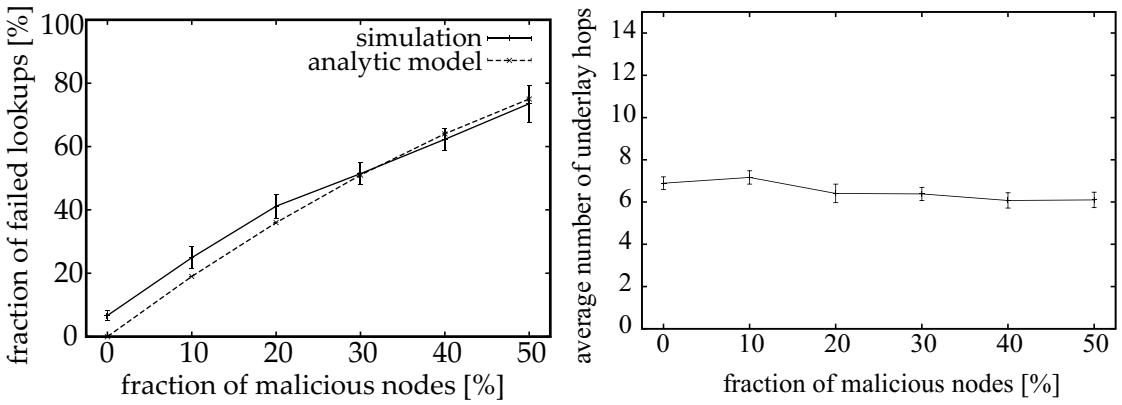
Attack and *Storage and Retrieval Attack* according to our first evaluation goal. As a result, intermediate and root nodes behave maliciously and drop lookup messages or reply faulty objects, respectively. Due to this fact, we assume a reduced fraction of successfully performed lookups (σ_C) as predicted by Equation 5.7. This fraction of successfully performed lookups is a function of the average number of required overlay hops (h) and the fraction of malicious nodes (f). However, the average number of overlay hops can be assumed as predicted in Equation 5.6. We will validate this analytic model according to our second goal, in the following evaluation.

$$\sigma_C = (1 - f)^h \tag{5.7}$$

The combined attack was further evaluated by a wired testbed using Pastry [35]. This testbed evaluation of Pastry has been harnessed to analyze the effect of these attacks on the proximity metric. Therefore, a *FreePastry* implementation has been deployed on two different testbeds. Those two testbeds, PlanetLab and G-Lab, strongly differ according to their characteristics, as discussed earlier.

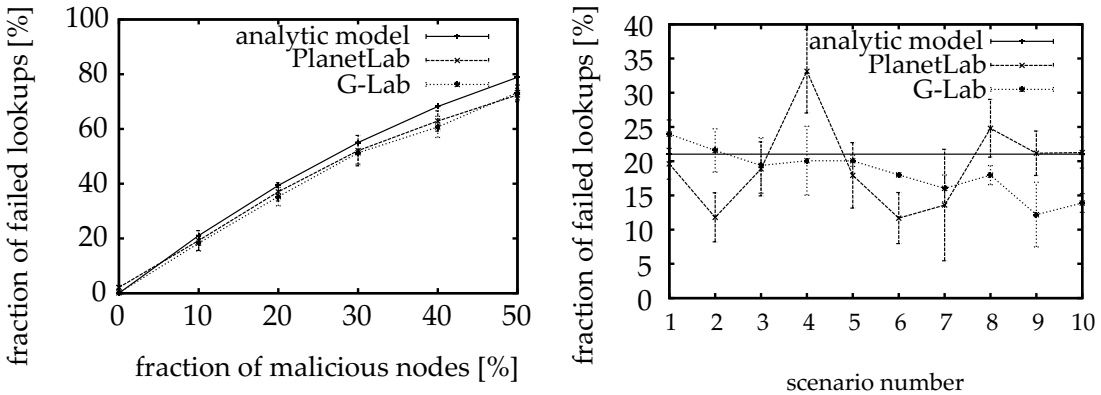
In order to evaluate the combined attack, 100 nodes have been selected from each testbed randomly and 5 virtual Pastry peers have been set up on each physical testbed node. Thus, an overall network size of 500 nodes per testbed could be achieved. The fraction of maliciously behaving peers had been varied from 0% up to 50% of the overall number of peers and each setting has been evaluated for 30 minutes. For this experiment, 10 iterations per setting were used and, furthermore, the maliciously behaving peers were changed each time, thus equalizing the occasionally strong impact of single peers as described in the following paragraphs. The testbed based evaluation is used to identify vulnerabilities of Pastry's algorithms as required by our last evaluation goal.

The evaluation of the combined attack is based on two metrics, the fraction of failed lookups (f_{lookup}) and the average number of underlay hops (T_{hops}), to analyze the attack. Furthermore, default parameters as defined in Chapter 4.1.4 are used.



(a) Fraction of failed lookups (b) Number of underlay hops

Figure 35: Evaluation results of the combined attack on a Clustered Pastry system



(a) Fraction of failed lookups (b) Impact of boot order

Figure 36: Results of the testbed analysis of the combined attack

RESULTS OF THE EVALUATION OF COMBINED ATTACKS

The overall fraction of failed lookups (f_{lookup}) of the combined attack is strongly increased when compared to the outcomes of the evaluations of the underlying attacks, as predicted by the analytic model. As shown in Figure 35a, even a small fraction of malicious nodes is able to strongly affect the lookup mechanism of the Clustered Pastry system. As assumed earlier, the average number of underlay hops (T_{hops}) is comparable to scenarios that are under attack of the *Incorrect Lookup Routing Attack* only as shown in Figure 35b.

TESTBED ANALYSIS

The averaged results of both testbeds are very similar as shown in Figure 36a. As Equation 5.4 provides only an upper bound for the hop length, the theoretical packet delivery rate predicted by Equation 5.3 is lower compared to the testbed results. The gap between the averaged testbed and theoretical results increases further when the fraction of malicious nodes increases. Yet, the average results of the testbeds are

reasonably close to the predicted results by the mathematical model proposed by Castro et al. [17].

On the other hand, analyzing single scenarios reveals effects that are not considered in that mathematical model. We discuss the scenario with 10% malicious peers in which we analyzed the impact of the distribution of malicious peers with respect to the order with which they join the DHT. In Figure 36b the packet loss ratio is shown according to the setting number. In the first setting, the first 50 nodes that join the network behave maliciously (nodes 1 to 50 in the boot order). In the second scenario, nodes 51 to 100 behave maliciously and so on. The results of the two testbeds vary significantly. As nodes in PlanetLab differ strongly in network load and link quality, Pastry's Round Trip Time (RTT) based proximity metric that determines which nodes are listed in the *routing table* strongly affects the impact of the routing attack: malicious peers with a good connection are included in *routing tables* with a higher probability than benign peers with an average connection. However, in G-Lab, nodes are homogenous regarding load and connectivity. Therefore, the RTT is mostly equal. This results in an increased impact of malicious nodes which boot first during the scenario. As those nodes were distributed in the *routing tables* at the beginning and are not replaced by nodes that provide a better RTT, their impact on the lookup process is increased.

The proximity metrics influence the distributions of links to nodes in the *routing tables*. Yet, this can be exploited in order to increase the impact of routing attacks. For example, a maliciously behaving node that plans to attack a specific node may position itself geographically close in order to provide the best RTT. As a result, this malicious node would be represented with a high probability in the victims *routing table* and, therefore, is able to attack this node. Due to this reason and as a proximity metric would result in an increased traffic overhead (see also Chapter 3.5.3), no proximity metric has been used in context of Clustered Pastry.

SUMMARY OF THE COMBINED ATTACK

To sum it up, we have shown in the previous paragraphs the impact of a attack that combines the *Storage and Retrieval Attack* with the *Incorrect Lookup Routing Attack* as required by our first evaluation goal. By combining those attacks, we were able to increase the overall impact of the attack on the availability of services provided to the Clustered Pastry system. Therefore, security mechanisms are required that are able to collaborate in order to ensure the robustness of our Clustered Pastry system. Moreover, we were able to validate the analytic model of the combined attack as required by our second evaluation goal.

Furthermore, we harnessed a testbed evaluation of the combined attack to detect vulnerabilities introduced by the overlay as required due to our third evaluation goal. As a result, we have shown that the proximity metric strongly affects the impact of the combined attack. Moreover, this metric can be exploited by maliciously behaving nodes in order to increase the impact of their attacks. Therefore, no proximity metric is used in the context of Clustered Pastry.

5.4 CHAPTER SUMMARY

After we have designed and evaluated the Clustered Pastry system in the previous two chapters, security threats for the resulting MP2P system have been discussed in this chapter. Therefore, existing attacks and security mechanisms on the underlying architectures have been surveyed.

Security mechanisms, which have been proposed for MANETs, were mostly not designed for a specific application. Therefore, we assume that the underlay of our MP2P system can be secured by using these existing mechanisms. However, mechanisms developed for DHTs often require either a reliable transmission rate or an increased amount of bandwidth. Due to this fact, we compared challenges introduced by the overlay with the related work in the area of MP2P security. As a result, we were able to identify open challenges in the area of routing attacks on the overlay's lookup functionality.

Based on this survey, three challenging attacks on the Clustered Pastry's overlay have been discussed and evaluated by an analytical approach, by simulation, and, in case of a combined attack, based on testbed results. We further harnessed the results simulation based evaluation in order to validate the analytic models. All of those analyzed attacks are capable to affect the availability of the networks services in terms of denying stored objects. Furthermore, the negative affect of a combined attack on the proximity metric of a DHT has been validated.

To sum it up, security threats introduced by the overlay routing have been discussed and analyzed. As a result, we were able to show that routing attacks and maliciously behaving root nodes have a high impact on MP2P systems as our Clustered Pastry. This impact on the network can even be increased by combining two attacks. As a result, security mechanisms are highly required to provide robustness against those attacks and to ensure reliable services of MP2P systems.

6

SECURITY MECHANISMS FOR MOBILE PEER-TO-PEER ARCHITECTURES

»Security is about preventing adverse consequences from the intentional and unwarranted actions of others.«

— Bruce Schneier

In Chapter 3 Clustered Pastry, a mobile, wireless system has been introduced that is able to store data in a decentralized way. By using the location awareness of the participating nodes during routing, services can be provided efficiently and in a reliable way by this clustered approach.

The characteristics of this system introduce vulnerabilities that can be exploited in order to attack the Clustered Pastry system as discussed in the previous chapter. This includes routing attacks as well as malicious behavior of root nodes. Therefore, three prominent and challenging attacks have been discussed extensively in the previous chapter. In order to ensure a reliable operation of the Clustered Pastry system, security mechanisms are required to provide robustness against those attacks.

To provide robustness to the *Storage and Retrieval Attack*, replication mechanisms are discussed in the first part of this chapter. However, MP2P systems benefit from replicas beyond security issues. Thus, we survey existing replication mechanisms that have been proposed for MANET and P2P systems in the light of MP2P scenarios and discuss their shortcomings in this chapter. Moreover, we introduce replication mechanisms that have been adapted to the characteristics of our Clustered Pastry system in the context of this thesis. In the second part of this chapter, security mechanisms are discussed that ensures that lookup requests are not dropped or misrouted by maliciously behaving intermediate nodes. Thus, two mechanisms that have been proposed for DHTs are analyzed in MP2P scenarios. Moreover, our *Overlay WatchDog* [36] is proposed to provide robustness against the *Incorrect Lookup Routing Attack*. Therefore, the characteristics of a wireless transmission are exploited to detect maliciously behaving nodes. The last part of this chapter discusses validation mechanisms to detect Forged Reply messages. Thus, two new validation mechanisms are developed [40] and compared to an approach that has been proposed by Castro et al. [17].

6.1 REPLICATION MECHANISM

MP2P systems, as assumed in this thesis provide services in terms of the storage and retrieval of data objects. The *availability* of these services is essential as discussed in the previous chapter. However, even when considering a benign behavior of all participating nodes during the routing of a root node, those services may be unavailable due to multiple reasons as long as these objects are stored at a single node only. For example, a root node may deny the object or may have left the network. Therefore, a replication mechanism is required in order to increase the availability of

the stored objects in the network. In the following paragraphs, we survey challenges that can only be addressed by replication.

ENSURE BASIC FUNCTIONALITY

Objects may get lost due to churn in P2P [116] and, even more, in Clustered Pastry systems as additional churn has to be considered due to nodes that change the cluster. Whenever a node leaves the network unexpectedly, e.g., due to node failure, objects maintained by this node may become unavailable [115]. Therefore, a basic replication mechanism has already been introduced in Chapter 3.6.1.

In addition, replicas can be used to increase the probability that a lookup is performed successfully and, furthermore, reduce the overhead generated due to the retrieval of an object. By distributing replicas in the network, each node that initiates a lookup is able to retrieve the requested object from the root of a replica that is located geographically closest.

PROVISION OF ROBUSTNESS AGAINST ATTACKS

As discussed in the previous chapter, the *availability* of objects can be strongly affected by maliciously behaving nodes. Distributing replicas in the network can be harnessed to increase the robustness against multiple attacks including routing attacks and, in particular, the *Storage and Retrieval Attack*. Even though other mechanisms can be used to decrease the impact of routing attacks, only replicas can be used to improve the network's robustness against maliciously behaving root nodes.

OBJECT AVAILABILITY IN PARTITIONED NETWORKS

In MP2P scenarios, a single node or a group of nodes may be temporarily disconnected from the rest of the network. This may be a result of the nodes limited transmission range, the mobility, and the resulting highly dynamic structure of the network. Due to this fact, objects that are solely stored at those disconnected nodes are no longer available to the rest of the network. By harnessing replicas, data loss as a result of network partitioning can be reduced.

6.1.1 *Challenges in Mobile Peer-to-Peer Systems*

We assume that several challenges have to be considered when developing a replication mechanism due to the characteristics introduced by the MP2P system. Once again, the strongly limited resources have to be taken into account, in particular the available bandwidth. Due to this fact, the number of replicas as well as the distribution mechanism has to be designed carefully. Storing only few replicas may result in a low robustness to maliciously behaving nodes. Yet, a high number of replicas results in a high traffic overhead due to uploading and updating those replicas. Earlier attempts to distribute replicas in MANETs further assumed that the mobile devices are strongly limited in their local memory [12]. Yet, we consider only small scale objects in the context of this thesis as defined by our disaster relief scenario (see also Chapter 3.1). Furthermore, most modern mobile devices provide a large memory nowadays (often more than 1 GB). Therefore, storage limitations are neglected in the context of this thesis. Besides limited resources, the dynamic topology introduces challenges for the replication mechanism [79]. Objects have to be redistributed frequently due

to cluster-based churn. Furthermore, objects may be unavailable due to network partitioning [79].

To sum it up, a replication mechanism has to balance the costs of replication regarding bandwidth with the resulting benefits as the increased robustness of the network.

EXISTING REPLICATION MECHANISMS IN THE LIGHT OF MOBILE PEER-TO-PEER

As discussed in Chapter 2, several replication mechanisms have been developed for MANETs and P2P systems. Even though those mechanisms can be deployed in MP2P scenarios as well, harnessing Clustered Pastry's structure to increase the benefits for the replication of objects is assumed to be more promising as we are able distribute the replicas more efficiently.

Replication mechanisms that were introduced for DHTs are mostly highly structured due to the characteristics of the overlay [48][10][90]. Those mechanisms harness the virtual identities of objects to replicate them on a specific set of nodes. Due to this fact, objects can be stored, retrieved, and updated at root nodes with a low effort in terms of maintenance costs. For example, replication mechanisms proposed by Rowstron et al. [89] and Castro et al. [17] store replicas at the virtual neighbors of the root node of an object. Though, most overlay based replication mechanism does not consider the geographical location of the root nodes during the replica allocation. As mentioned earlier, this may be essential to increase the availability of objects under the light of network partitioning. Furthermore, when considering the geographical location of nodes during the distribution of the replicas, the number of required messages can be reduced.

On the other hand, mechanisms developed for the MANET underlay are often based on the geographical distribution of objects [46][45][47][69]. However, those mechanisms are, in most cases, uncoordinated and use a flooding based allocation mechanism to distribute or update the replicas. Yet, only few coordinated allocation mechanisms have been introduced in related work. A coordination mechanism is required to organize the replicas in the MANET. However, this mechanism introduces traffic overhead that should be avoided in the context of MP2P systems.

By combining the benefits of MANET and DHT replication mechanisms, we may be able to overcome the previously discussed challenges. As a result, we propose a structured and location aware replication mechanism that has been developed for the Clustered Pastry system.

6.1.2 *Basic Replication*

Clustered Pastry uses a basic replication mechanism in order to avoid data loss due to churn as already discussed in Chapter 3.6.1. The basic replicas (hereinafter to be referred to as local replicas) are locally replicated at the virtual neighbors of the root node as shown in Figure 37. These virtual neighbors are located within the same cluster due to the Clustered Pastry's structure.

However, those objects are not used during routing but only to redistribute the objects when nodes leave a cluster. Therefore, they cannot be used to provide robustness against maliciously behaving nodes. Furthermore, the replicas are stored within the

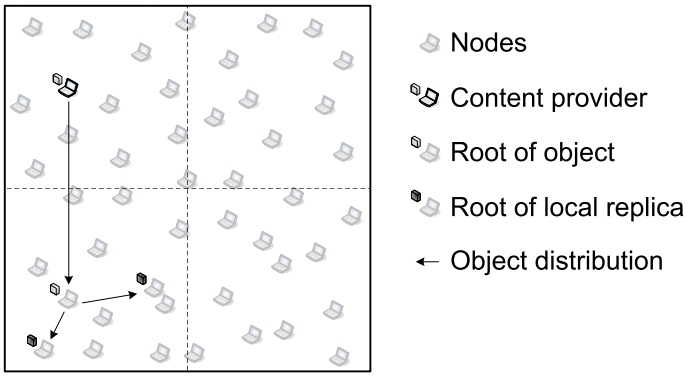


Figure 37: An example for the distributing secondary replicas on virtual neighbors

same cluster at virtual neighbors. Due to this fact, local replicas are located in the same geographical area. As a result, we assume that this basic replication mechanism offers only a limited robustness against network partitioning as all replicas may become unavailable when a single node group is disconnected from the Clustered Pastry network.

6.1.3 *Inter-Cluster Replication*

Our Clustered Pastry system uses clusters to provide a location-aware distribution of the overlay identifier as discussed in Chapter 3. This awareness of the geographical location of the nodes can further be harnessed to optimize the distribution of replicas. Thus, we developed the Inter-Cluster Replication (ICR) mechanism.

Three major requirements have been previously identified that have to be satisfied by an MP2P replication mechanism. The basic replication scheme ensures the availability of stored objects even during churn. The remaining goals, which include the robustness against maliciously behaving root nodes and network partitioning, have to be met by the adapted replica mechanism that is discussed in the following paragraphs.

We propose to distribute the replicas among the clusters. For example, in a setting with four clusters, a replica is stored at each cluster as shown in Figure 38. Thus, this approach is based on the geographically separation of clusters. As described by Equation 6.1, the identifier of the replicas ($ID_{replica}$) can be easily determined as long as the identifier of the requested object (ID_{object}) and the number of global replicas (k) is known. Furthermore, the highest identifier in the identifier space (ID_{max}) is used to derive the identifiers of the stored objects. However, the number of global replicas including the original object ($k+1$) has to be smaller than the overall number of clusters. Whenever a higher ratio of replicas stored in a single cluster is required, the number of local replicas has to be increased. Furthermore, k is defined as a non-negative integer to the power of 2.

$$ID^{j}_{replica} = (ID_{object} + \frac{(ID_{max}+1)*j}{k+1}) \mathrm{mod} (ID_{max}+1),\ j = 1,..,k \qquad (6.1)$$

Due to this distribution mechanism, we can assure that replicas are stored at geographically separated nodes. Therefore, we assume a high availability of objects

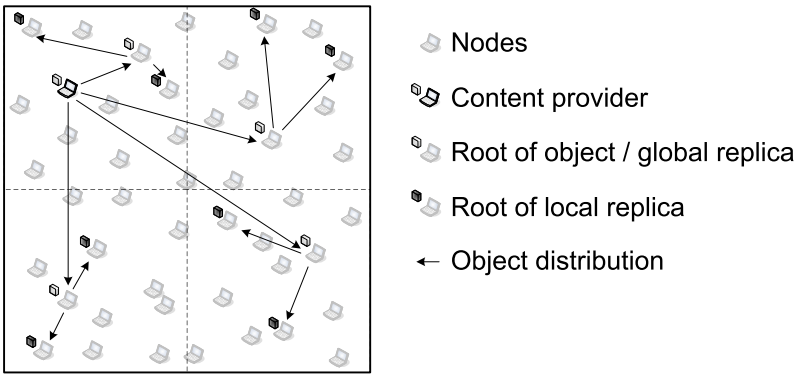


Figure 38: An example for an *Inter Cluster Replication*

even when the network is split due to, e.g., node mobility. As a result, this approach provides a high degree of robustness against data loss due to network partitioning. Furthermore, distributing nodes equally in the deployment area increases the robustness against maliciously behaving nodes. As mentioned earlier, Sybil attacks can increase the impact of, e.g., *Storage and Retrieval Attacks* as a single malicious node is able to generate multiple virtual identities. Though, when replicas are used, a malicious node has to get hold of each replica of the object it wants to deny. As the replicas are distributed in different separated clusters when using the ICR mechanism, a Sybil attack cannot be used in order to obtain each replica as the physical Sybil node and, therefore, each virtual identity is located in a single cluster. However, threshold areas can be exploited in order to generate virtual identities in adjacent clusters.

This replication mechanism is further similar to the approach introduced by Harvesf and Blough [48][49]. Yet, this approach only distributes the replicas equally in the namespace of the overlay identifier and not in the deployment area.

This replication mechanism further results in an increased traffic overhead due to uploading or updating objects as the replicas are stored at geographically separated parts of the network. Furthermore, the overall number of clusters limits the maximum number of replicas as there never should be more than a single global replica of the same object within a cluster.

STORAGE OF OBJECTS

Whenever an object has to be stored in the network, the object identifier has to be determined as already discussed in Chapter 3.5.4. Thereafter, the identifiers of the global replicas are derived based on the objects identifier. The ICR mechanism uses the object provider to distribute the object and the global replicas in the network. However, if a lookup for a root node fails, no global replica will be stored in this cluster. Subsequently, local replicas are distributed locally in the cluster by the root nodes of the object and the global replicas. Those local replicas are required in order to ensure that neither objects nor replicas get lost due to churn. Considering this distribution algorithm, the overall number of global replicas including the original

object (k_{real}) depends on the fraction of successful lookups (σ) and can be derived as shown in Equation 6.2.

$$k_{real} = \sum_{j=1}^{k} \sigma_j \tag{6.2}$$

RETRIEVAL OF OBJECTS

When requesting an object, the source of the lookup determines the geographically closest replica. Thereafter, a lookup is initiated and, after receiving the appropriate reply message, the requested object is retrieved. Yet, whenever no reply message or object is received or when a faulty object is retrieved, a lookup for another replica is initiated. This procedure is repeated until either the object is retrieved successfully or all root nodes of global replicas have been contacted.

The fraction of successfully retrieved objects (p) is a function of the number of distributed objects in the network (k_{real}), the fraction of successful lookups (σ) and the fraction of malicious nodes that performs *Storage and Retrieval Attacks* in the network (f) as shown in Equation 6.3. Routing attacks as the *Incorrect Lookup Routing Attack* are not directly considered in this equation but are included in the fraction of successful lookups (σ).

$$p = 1 - (1 - \sigma(1 - f))^{k_{real}} \tag{6.3}$$

DISCUSSION

By harnessing this replication mechanism, replicas are distributed equally in the deployment area. This mechanism distributes an average number of overall replicas ($r_{overall}$) that is based on the number of correctly allocated global replicas k_{real} and the number of local replicas per global replica (r_{local}) as shown in Equation 6.4.

$$r_{overall} = k_{real} * r_{local} \tag{6.4}$$

The ICR mechanism introduces multiple benefits. Due to the equally distributed replicas, the robustness against data loss due to network partitioning and *Storage and Retrieval Attacks* is provided (see also Chapter 6.2.2 and Chapter 6.2.4). Furthermore, the fraction of successfully received objects can be improved due to the replicas. Moreover, overhead due looking up an object can be reduced by retrieving objects by a geographically close replica. However, distributing and updating these replicas introduces overhead in terms of traffic. Furthermore, a high amount of traffic is generated locally whenever a node shares an object. This may effect the efficiency of the ICR replication mechanism.

6.1.4 *Cyclic Replica Allocation*

The ICR mechanism introduces multiple benefits to a location aware MP2P system as Clustered Pastry. However, ICR's replica distribution mechanism of the is not optimal. A high amount of traffic is introduced locally by the content provider. This node has to centrally lookup the root of the original object as well as the root nodes of the global

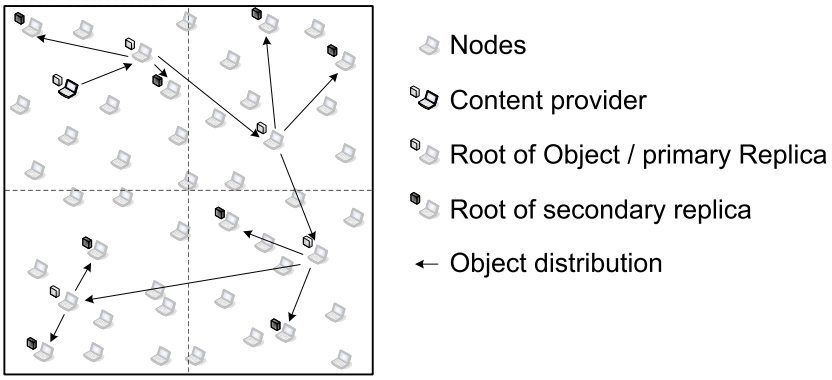


Figure 39: An example for the *Cyclic Replica Allocation* scheme

replicas. Thereafter, all of these objects and replicas have to be uploaded by the content provider. Furthermore, overhead is generated as objects have to be transmitted to distant nodes. The Cyclic Replica Allocation (CRA) harnesses a different distribution scheme. The content provider uploads the object to the geographically closest root node only. Thereafter, this object is forwarded by this root node to another logically close root node. As a result, objects are not distributed by a single node but by a set of root nodes.

The CRA shares characteristics with the ICR, which has been discussed in the previous section. Both replication mechanisms benefit from the location-dependent overlay identifier. Furthermore, both mechanisms distribute the replicas equally in the identifier namespace and harness the same retrieval algorithm. Yet, they differ with reference to the replica allocation scheme.

STORAGE OF OBJECTS

The CRA does not differentiate between the original object and the global replicas. Due to this fact, both, the global replicas and the original object are hereinafter referred to as global replica. Whenever, an object has to be stored in the Clustered Pastry system, the set of global replicas is determined as shown in Equation 6.1. Based on this set, a replication order is defined. Therefore, Pastry's *ring* structure is harnessed as discussed in Chapter 2.1.3. The global replicas are ordered according to their virtual clockwise distance to the content provider on this *ring* structure. However, a global replica that is located in the same cluster as the content provider is always prioritized in the replication order regardless of the virtual distance between those nodes.

After defining the replication order, the content provider initiates a lookup process for the root node of the first global replica. Thereafter, the new object is transmitted to the root node and is stored locally. This root node distributes local replicas as discussed in Chapter 6.1.2 and initiates a lookup process for the second root node according to the replication order. The object is forwarded by each root node until the last root node receives the object as shown in Figure 39.

This distribution scheme affects the overall number of stored global replicas including the original object (k_{real}^{CRA}) as described in Equation 6.5. Whenever a lookup process for a root node fails, also all subsequent replicas cannot be allocated due to the recursive propagation of global replicas. However, considering a fraction of 95%

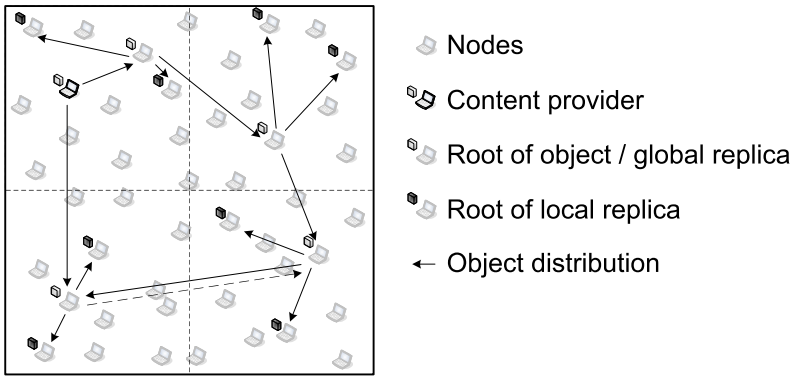


Figure 40: An example for the *Optimized Cyclic Replica Allocation*

of successfully performed lookups in a setting with 4 replicas, more than 3.5 replicas are actually stored in the network on average.

$$k_{real}^{CRA} = \sum_{j=1}^{k} \sigma^j \tag{6.5}$$

DISCUSSION

The CRA introduces an improved distribution mechanism and ensures that replicas are distributed efficiently with a high probability when considering a network with a reasonable fraction of successful lookup operations. Furthermore, the traffic of an upload is not anymore generated by a single node but is divided among a set of nodes, which are distributed in the deployment area. This may be essential especially when considering that some few nodes may provide a large fraction of the networks content. Moreover, as the global replicas are not distributed by a single node but by several root nodes and due to the dedicated replication order, the average distance in terms of underlay hops between the distributing node and the target root node can be reduced.

Yet, maliciously behaving nodes may further not be willing to forward an object correctly to the root node. As a result, the number of correctly distributed replicas in the network is strongly decreased as shown in Equation 6.6. For example, a fraction of 25% maliciously behaving nodes would reduce the number of available replicas by 50% on average in a setting with 4 global replicas.

$$k_{real-mal}^{CRA} = \sum_{j=1}^{k} (\sigma_j^j * (1-f)^j) \tag{6.6}$$

6.1.5 *Optimized Cyclic Replica Allocation*

The CRA mechanism introduces a vulnerability to malicious behavior as discussed in the previous paragraph. Therefore, the Optimized Cyclic Replica Allocation (OCRA) mechanism has been developed to overcome this weakness. By using two different replication paths to distribute the global replicas, the average fraction of successfully

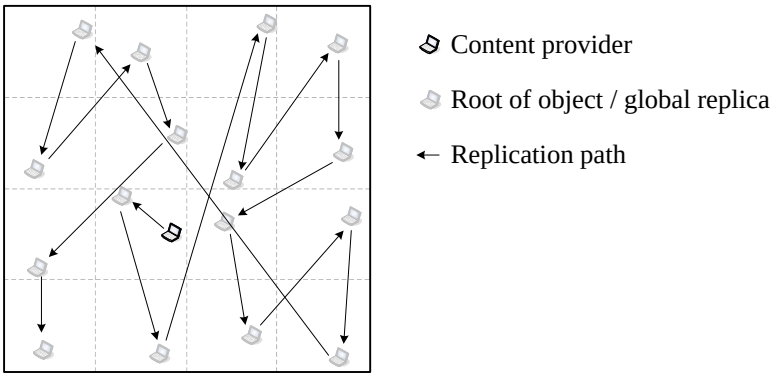


Figure 41: An example for the *Cyclic Replica Allocation* in a scenario with 16 clusters

distributed replicas can be strongly improved. Even when considering malicious behavior, a reliable distribution of objects can be assured.

OCRA, CRA and ICR harness the same algorithm to select the root nodes and to retrieve replicas. Yet, they differ on how the replicas are distributed in the network.

STORAGE OF OBJECTS

As shown in Figure 40, one copy of the object is forwarded to the root node that is located in the same cluster. A second copy of the object is forwarded to the next root node that is located counter clockwise in the virtual name space. Thereafter, both root nodes that received the object store it locally as global replica and forward this object. Therefore, those objects are forwarded clockwise and counter clockwise, respectively. As a result, redundancy is introduced due to the two replication paths. However, an increased overhead is introduced as at least one redundant lookup message has to be sent (indicated by the dotted lines in Figure 40).

Due to the OCRA mechanism, the number of successfully distributed replicas (k_{real}^{OCRA}) can be increased. As a result of the redundant replication paths, an increased fraction of replicas can be correctly distributed as shown in Equation 6.7.

$$k_{real}^{OCRA} = \sum_{j=1}^{k} \left[\sigma^j + (1-\sigma^j) * \sigma^{1+k-j}\right] \tag{6.7}$$

DISCUSSION

The OCRA mechanism is able to improve the availability of replicas in an MP2P network as shown in Equation 6.8. Furthermore, the allocation of objects is more reliable than either the ICR or CRA mechanism when considering a setting without maliciously behaving nodes (considering a scenario with 4 replicas and a fraction of successfully uploaded replicas of 95%). However, an increased number of sent lookup messages are introduced by the OCRA mechanism and, therefore, an increased amount of traffic overhead is generated due to the distribution of the global replicas.

$$k_{real-mal}^{OCRA} = \sum_{j=1}^{k} \left[\sigma^j * (1-f)^j + (1-\sigma^j) * \sigma^{1+k-j} * (1-f)^{1+k-j}\right] \tag{6.8}$$

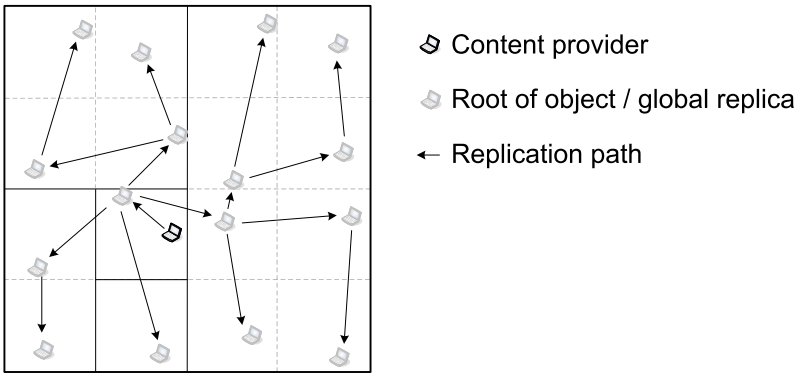


Figure 42: An example for the *Delegate Replica Allocation*

6.1.6 *Delegate Replica Allocation*

The previously discussed CRA mechanisms provide efficient services as long as settings with a limited amount of clusters are assumed. Yet, considering settings with 16 or more clusters, the cyclic distribution mechanisms replicate objects in an inefficient way. Global replicas are distributed on a zigzag route due to the structure of the clusters in the network and as a result of the CRA and the OCRA distribution scheme, as shown in Figure 41. As a result, an overhead is introduced due to the resulting replication paths.

We thus propose the Delegate Replica Allocation (DRA) mechanism to overcome this challenge and to distribute replicas more efficiently in a network with a high number of clusters. Therefore, aspects of the cluster allocation and the structure of the routing table of Clustered Pastry have been taken into consideration during the development of this replication mechanism. As a result, a highly structured dissemination of replicas can be achieved.

The DRA mechanism shares several characteristics of the previously introduced replication mechanisms. Thus, objects can be retrieved in a similar way. Yet, those mechanisms differ on how to determine the identifier of the next replica ($ID_{replica}$) that has to be stored. This identifier is a function objects identifier (id_{object}) and the cluster prefix of the given target cluster (id_{prefix}). Furthermore, a simple bit concatenation is required as shown in Equation 6.9

$$ID^{DRA}_{replica} = id_{prefix}, id_{object}[l - l_{prefix} - 1 : 0] \tag{6.9}$$

STORAGE OF OBJECTS

Whenever an object has to be stored in the network, the content provider determines the overlay identifier of all root nodes. Thereafter, the object is forwarded to the root node that is either located in the same cluster or that is the virtually close to the content provider. This first root node compares the list of replicas with the entries of its *routing table*. As a result, a lookup for each root node is initiated where a routing table entry is available that link to a node that is located in the same cluster. After determining the second set of root nodes, a copy of the object is transmitted. Afterwards, those root nodes use their routing table to determine the third set of root nodes locally. Therefore, the routing table entries are used but entries that differ on

the first prefix are neglected. The replicas are distributed in this way until all root nodes, which could be determined by a lookup, received a replica as shown in Figure 42. Moreover, with each subsequent set of root nodes, the number of neglected prefix digits is increased.

As a result, the number of correctly allocated replicas depends on the structure of Clustered Pastry and, therefore, on the number of clusters (C). The resulting number of replicas is shown in Equation 6.10.

$$k_{real}^{DRA} = \sigma * (1 + \sigma)^{(log_2(C))} \tag{6.10}$$

DISCUSSION

The DRA algorithm harnesses the structure of Clustered Pastry in order to distribute the replicas in a more efficient way. Therefore, the average number of underlay hops that is required to distribute replicas can be reduced. Moreover, this results in a increased fraction of correctly allocated replicas. However, this also results in a highly specialized replication mechanism. Thus, DRA can only be used efficiently in the context of the Clustered Pastry system.

6.1.7 *Conclusions on the Replication Mechanisms*

As discussed in this section, replicas are required in MP2P systems due to multiple reasons. Replicas can be harnessed to prevent churn and to reduce the impact of a network partitioning. Furthermore, a replication mechanism can be used to increase the networks robustness to maliciously behaving root nodes. In the previous paragraphs, we introduced and discussed five replication mechanisms.

The basic replication mechanism ensures the availability of objects even when nodes leave the network unexpectedly by distributing local replicas in the cluster of the object's root node. As a result, this mechanism provides robustness against churn. This basic mechanism is furthermore included in each of the other replication mechanisms. The four other replication mechanisms distribute global replicas equally in the identifier namespace and, due to the structure of Clustered Pastry, in the deployment areas. As a result, objects are only lost due to a network partitioning when multiple nodes that are located in different parts of the deployment area disconnect at the same time from the network. Moreover, robustness against maliciously behaving root nodes is introduced by those replication mechanisms (as will be proved in the next section).

The remaining four replication mechanisms share multiple characteristics but differ on the distribution mechanism. ICR uses the content provider to distribute the replicas while CRA distributes the replicas via the root nodes. OCRA is based on a similar algorithm as CRA but introduces redundancy in order to increase the number of correctly allocated replicas. Moreover, we developed the DRA to harness the structure of Clustered Pastry to optimize the distribution mechanism.

6.2 EVALUATION OF THE REPLICATION MECHANISMS

Besides the basic replication mechanism, four replication mechanisms have been introduced in the previous section. These mechanisms differ in their replica alloca-

tion algorithms. In the following paragraphs, the efficiency of those mechanisms is evaluated by means of simulation.

6.2.1 *Evaluation Goals, Metrics, and Methods*

In the following paragraphs, we discuss the goals of the evaluation. Moreover, we define the metrics that are harnessed to determine the efficiency of the replication mechanisms and introduce the evaluation tools.

EVALUATION GOALS

We harness replicas in order to increase the robustness against maliciously behaving root nodes and to minimize data loss due to unavailable nodes. Thus, the goals of the following evaluation are based on those two areas of application.

We introduced multiple replication mechanisms in the previous section. Thus, we have to evaluate the compare the efficiency of those mechanisms, as our first evaluation goal. This also includes the robustness provided by these mechanisms to maliciously behaving root nodes. Moreover, we introduced DRA as a replication mechanism, that is adapted to the structure of Clustered Pastry. Therefore, we have to compare the efficiency of the allocation algorithm of DRA with a non-adapted mechanism as our second evaluation goal.

Due to node mobility, groups of nodes may get disconnected from the MP2P network especially when considering sparse networks in terms of networks with a low node coverage. As a result, the availability of objects that are stored at these nodes may be affected. However, we assume that replicas can be used to increase the networks robustness against network partitioning. Thus, we have to validate the influence of replicas in networks with a low node density as our third evaluation goal.

EVALUATION METRICS

We use three evaluation metrics in order to evaluate the efficiency of the replication mechanisms: The fraction of failed lookups (f_{lookup}), the traffic introduced by the Clustered Pastry system (T) and the fraction of correctly allocated replicas ($f_{allocate}$).

The fraction of failed lookups (f_{lookup}) is used to indicate the availability of the data objects stored in the network. Therefore, we are able to determine the robustness of our replication mechanisms against attacks as the *Storage and Retrieval Attack* based on this metric. As we assume an increased traffic due to the distribution of the replicas, the overall traffic introduced by the Clustered Pastry system (T) is used as our second metric. As the bandwidth is limited due to the wireless underlay, the traffic overhead generated by the replication mechanisms has to be minimized. As the last metric, we harness the fraction of correctly allocated replicas ($f_{allocate}$) to indicate whether the replicas are distributed correctly in the network.

EVALUATION METHODS

We evaluate our replication mechanisms by the means of simulation. Therefore, we use our Clustered Pastry model that has been developed for the OMNeT++ simulator, which has been described in Chapter 4.1.3. Moreover, replication mechanisms as discussed in the previous section have been implemented. We assume default parameters of the Clustered Pastry model as defined in Chapter 4.1.4.

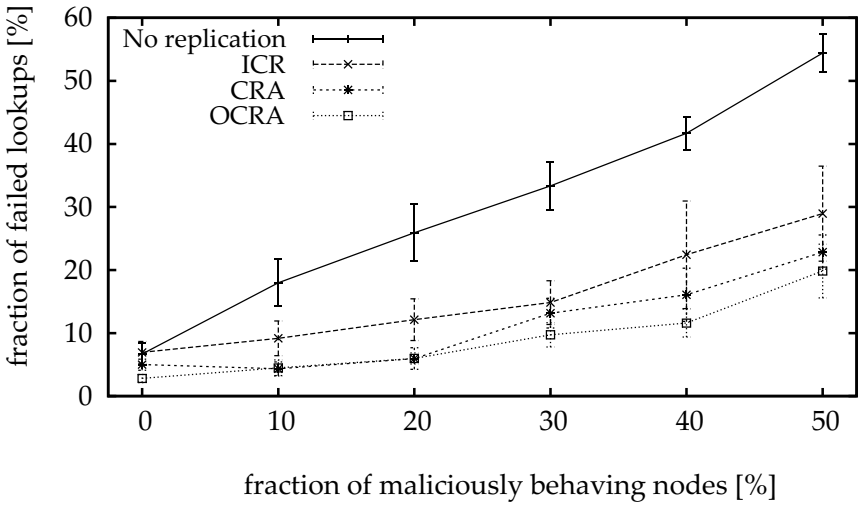


Figure 43: The impact of the *Storage and Retrieval Attack* on settings with and without replication mechanism

6.2.2 *Comparison of Replica Distribution Mechanisms*

In the previous section, multiple replication mechanisms have been introduced. Those mechanisms differ mostly on the how the replicas are distributed in the deployment area. When using the ICR mechanism, all replicas are distributed by the content provider, while the CRA and the OCRA only use the content provider to initiate the distribution of the replicas in the network. Thereafter, the distribution of the replicas is handled by the root nodes. We assume that this affects the overall traffic introduced by the Clustered Pastry system as well as the fraction of correctly allocated replicas.

However, in the following evaluation, we directly compare our replication mechanisms when under attack. Therefore, up to 50% of the root nodes behave maliciously and deny requested objects (*Storage and Retrieval Attack*). We use the three previously introduced metrics (f_{lookup}, T, $f_{allocate}$) to determine the efficiency of each replication mechanism.

RESULTS OF THE COMPARISON OF REPLICA DISTRIBUTION MECHANISMS

The replication mechanisms distribute a global replica in each of the clusters. Yet, those mechanisms differ in the way the replicas are allocated and, therefore, in the fraction of correctly allocated replicas ($f_{allocate}$). The ICR mechanism harness the content provider to distribute the replica directly. Thus, a large amount of traffic is generated by the content provider due to this mechanism. Moreover, the geographical distance between a root node and the content provider may be high. As a result, only 60% of the replicas are allocated in the network on average. The CRA mechanism has been developed to overcome this drawback. CRA harness the root nodes to distribute the replicas in the network. The content provider only has to forward the object to the geographically closest root node. Thereafter, the object is forwarded from this root node to another root node that is located in the adjacent cluster. The objects are forwarded by the root nodes until all replicas are correctly distributed. Thus, the objects always have only to be forwarded to the next cluster. Yet, when a single lookup fails, no further replicas can be distributed. However, the fraction of correctly

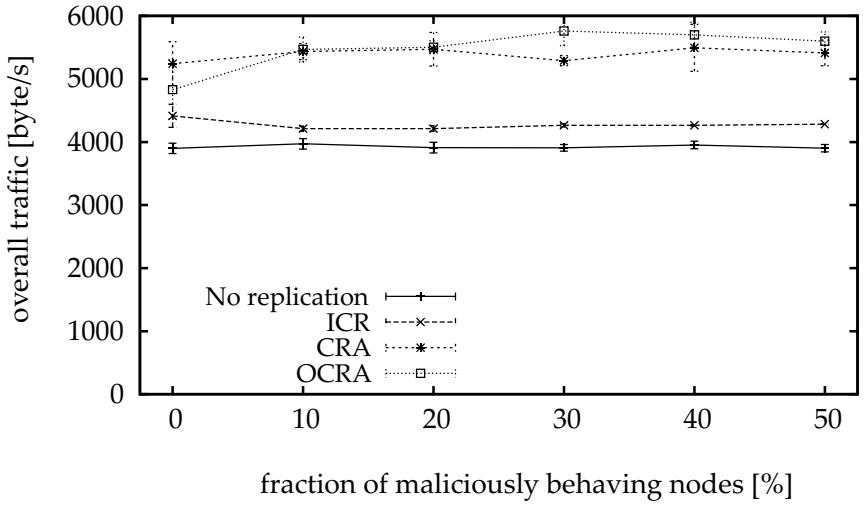


Figure 44: The impact of the *Storage and Retrieval Attack* on settings with and without replication mechanism

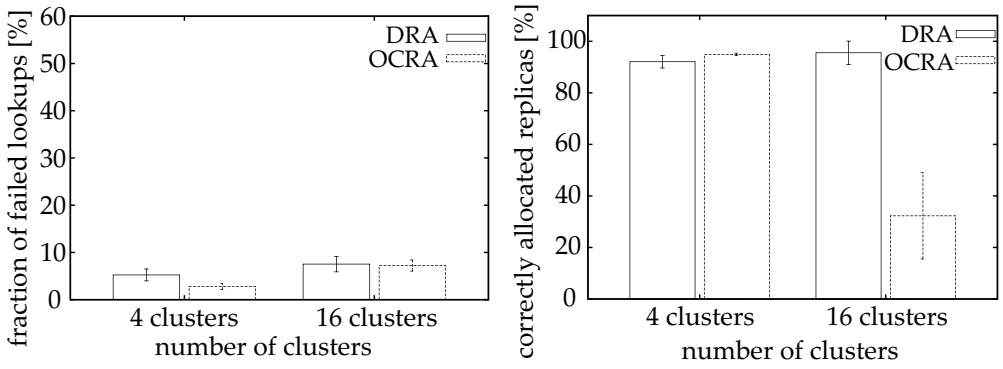
allocated replicas ($f_{allocate}$) introduced by the CRA mechanism is increased to 88%. Though, we were able to improve the fraction of correctly allocated replicas ($f_{allocate}$) by introducing a second replication path by the OCRA mechanism. As a result, we were able to ensure that 95% of all replicas were allocated correctly.

All of our proposed replication mechanisms are able to increase the robustness against the *Storage and Retrieval Attack*. The ICR and CRA mechanism are able to reduce the impact of those maliciously behaving nodes by providing replicas. Therefore, the fraction of failed lookups (f_{lookup}) can be strongly reduced as shown in Figure 43. However, as OCRA is able to allocate replicas in a more reliable way due to the redundancy introduced by the allocation algorithm, also the fraction of failed lookups can be slightly improved compared to results provided by ICR or CRA.

Yet, all of those replication mechanisms introduce an increased traffic (T). As we propose to distribute the replicas in different clusters, the overall traffic is increased by up to 40% as shown in Figure 44. ICR introduces the lowest traffic overhead of 10% due to the reduced number of correctly allocated replicas. CRA introduces a traffic, that is increased by nearly 38%. Due to the redundant allocation algorithm of OCRA a slightly higher traffic is introduced by this replication mechanism. On average the overall traffic of the Clustered Pastry system is increased by 40% when using OCRA mechanism compared to a setting without any replication mechanism.

SUMMARY OF THE COMPARISON OF REPLICA DISTRIBUTION MECHANISMS

ICR, CRA, and OCRA are all able to increase Clustered Pastry's robustness against maliciously behaving root nodes. Yet, ICR introduces only a low fraction of correctly allocated replicas ($f_{allocation}$) as discussed previously. Thus, CRA provides better results when considering the fraction of failed lookups (f_{lookup}) and the fraction of correctly allocated ($f_{allocation}$) replicas. However, by introducing redundancy to the replication mechanism, OCRA performs even better as CRA but on the cost of a slightly increased traffic.



(a) Fraction of failed lookups (b) Correctly allocated replicas

Figure 45: Comparison of 4 and 16 cluster settings with OCRA and DRA

6.2.3 *Replication in Settings with a High Number of Clusters*

Both, CRA and OCRA have been optimized for settings with either 2 or 4 clusters. As discussed in the previous section, we assume that the efficiency of those mechanisms is degraded in settings with an increased number of clusters. Yet, an increased number of clusters is required in large scale settings with a high number of participating nodes. Therefore, we developed the DRA mechanism, which harnesses the structure of Clustered Pastry to allocate the replicas. In the following paragraphs we evaluate this replication mechanisms in settings with 4 and 16 clusters and compare the outcomes to the results provided by the OCRA mechanism.

RESULTS FOR SETTINGS WITH A HIGH NUMBER OF CLUSTERS

Both replication mechanisms provide reliable results when considering the fraction of failed lookups (f_{lookup}) as shown in Figure 45a. Yet, when considering the results of settings with 16 clusters, the fraction of failed lookups is slightly increased compared to a settings without replication as a high traffic overhead is generated due to the high number of replicas that are distributed in a relatively small field.

Though, both replication mechanisms differ in the fraction of correctly allocated replicas ($f_{allocation}$) in setting with a high number of clusters. As shown in Figure 45b, OCRA is not able to allocate the replicas reliable when considering a setting with 16 clusters as a result of the distribution order of the replicas. However, as our DRA mechanism is adapted to the structure of Clustered Pastry, the replicas can be allocated in a far more reliable way. Thus, the average number of underlay hops that is required to allocate the replicas can be reduced and, therefore, replicas can be allocated with a higher probability.

Moreover, both replication mechanisms introduce the same traffic (T) due to the replication of objects in settings with 4 clusters. However, when increasing the number of clusters, DRA introduces a higher amount of traffic as a result of the higher amount of correctly replicated objects.

SUMMARY OF THE REPLICATION IN SETTINGS WITH A HIGH NUMBER OF CLUSTERS

Both, OCRA and DRA allocate replicas with a very high probability in settings with 4

Setting	Nodes	Deployment area	Connectivity
A	100	1100m x 1100m	99.2%
B	100	14000m x 1400m	94.5%
C	100	1600m x 1600m	87.5%
D	100	1700m x 1700m	80.5%
E	100	1800m x 1800m	72.5%

Table 6: Deployment area in settings with sparse networks

clusters. Yet, when considering an increased number of clusters, the OCRA mechanism fails to distribute replicas in an efficient way due to long replication paths. DRA is adapted to the structure of the Clustered Pastry system and, therefore, the average distance between the node that distributes the replica and the root node in terms of underlay hops is reduced. As a result, replicas are allocated in a more reliable way. However, in order to use DRA in combination with any other MP2P system, the allocation algorithm of DRA has to be adapted in order to ensure the efficiency of this approach.

6.2.4 *Replication Mechanisms in Sparse Settings*

MP2P systems as Clustered Pastry are based on a wireless, mobile underlay. Thus, single nodes or a group of nodes may be disconnected from the rest of the network due to the dynamic topology of the underlay. As a result, objects that were stored at the disconnected nodes are not anymore available for the rest of the network. By distributing replicas in the network, the availability of the objects can be improved as multiple root nodes have to disconnect at the same time from the network in order result in a lost object.

However, we assume that the specific allocation of the replicas affects the efficiency against data loss due to network partitioning. Replication mechanisms developed for DHTs often harness the overlay identifier to allocate replicas. Thus, the distribution mechanism proposed by Castro et al. [17] uses virtual neighbors of the root node to store the replicas is used as reference model. In the following paragraphs, we compare the impact of sparse settings with a replication mechanism that harness the virtual neighbors as proposed by Castro et al. with our OCRA mechanism. As shown in Chapter 6.2.2, OCRA provides the best results in settings with 4 clusters and considers the geographical position of the nodes during replication. Thus, objects are distributed equally in the deployment area.

Five settings were evaluated that differ in the size of the deployment area as shown in Table 6. While setting *A* is our default setting for a 100 node simulation as defined in Chapter 4.1.4, the subsequent settings are based on an increased size of the deployment area. We used the ANSim [50] tool to determine the resulting probability for two random nodes in the network to establish a connection between each other. As shown in the table, this connectivity strongly decreases when increasing the size of the deployment area. Moreover, each of the replication mechanisms stores four replicas in the network. In order to evaluate this scenario, all three of the previously mentioned metrics are used (f_{lookup}, T, $f_{allocate}$).

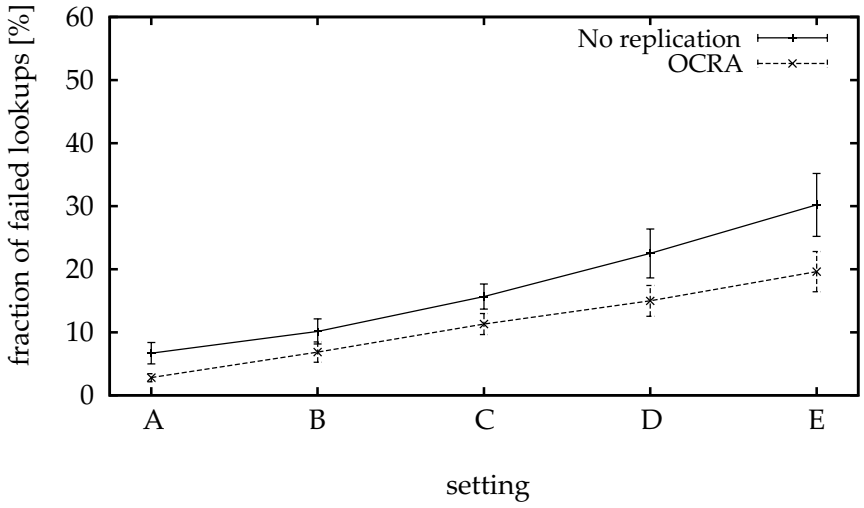


Figure 46: The impact of the *Storage and Retrieval Attack* on settings with and without replication mechanism, measured as the fraction of failed lookups

RESULTS FOR THE REPLICATION MECHANISMS IN SPARSE SETTINGS

When increasing the field size, the probability of a network partitioning increases. Thus, nodes or groups of nodes may be disconnected from the rest of the MP2P network. This results in an increased fraction of failed lookups (f_{lookup}) as shown in Figure 46. Settings that are based on the approach proposed by Castro et al. [17] introduce a higher vulnerability compared to our OCRA mechanism. As Castro's replication mechanism harnesses the virtual neighbors of the root node to replicate the object, those replicas are located in the same cluster and, therefore, in the same geographical area. As a result, the probability increases that all root nodes that provide a replica of the same object are disconnected from the network due to a network partitioning at the same time. Thus, while setting A provides a fraction of failed lookups of about 5%, this fraction is increased up to 30% in setting E. However, our proposed OCRA replication mechanism distributes the replicas equally in the deployment area. As a result, a higher robustness against data loss due to a network partitioning can be provided. Yet, even though OCRA provides a reduced fraction of failed lookups, the results of the settings clearly indicate that approximately 20% of all lookups fail in the setting E.

However, considering the fraction of correctly allocated replicas ($f_{allocate}$) by the OCRA, simulation results show a mostly static distribution rate of about 95% in the settings A to C. Therefore, we assume that most of the objects were correctly stored in the network. Based on this result, the increased fraction of failed lookups is mostly a result of disconnected nodes that initiates a lookup. Thus, nodes that are connected to the MP2P network are still able to retrieve the most of the objects correctly. Yet, in setting D and E the fraction of correctly allocated replicas is reduced to 85%. This indicates that an increased amount of nodes are disconnected from the MP2P network and, therefore, multiple objects may be unavailable.

OCRA introduces an overall traffic (T) that is increased by 24% when compared to the Castro et al.'s approach. This is a result of the structure of Clustered Pastry. OCRA stores replicas in each of the networks clusters and, therefore, in different geographical areas. Castro's replication mechanism stores the replicas at virtual neighbors of the

object's root node and, therefore in the same cluster. As a result, those objects are replicated in the same geographically area and introduce a reduced amount of traffic during the allocation of the replicas.

SUMMARY ON THE REPLICATION MECHANISMS IN SPARSE SETTINGS

In sparse networks, the availability of the services provided by Clustered Pastry is affected due to network partitioning. In the previous paragraphs, we have shown that replication mechanisms, which distribute the replicas equally in the deployment area, can be used in order to increase the availability of the objects stored in the network. Furthermore, we have shown, that our replication mechanism provides better results than a mechanism based on replicating objects at virtual neighbors as proposed by Castro et al. [17].

6.2.5 *Conclusions on the Evaluation of the Replication Mechanisms*

In this section, we evaluated the replication mechanisms proposed previously. Thus, we evaluated our proposed replication mechanisms in the light of maliciously behaving nodes performing the *Storage and Retrieval Attack* as required by our first evaluation goal. All of these mechanisms are able to improve the networks robustness against malicious root nodes. However, even though the OCRA replication mechanism introduces the highest traffic overhead, this mechanism also provides the best results in terms of the fraction of correctly allocated replicas ($f_{allocate}$) and the fraction of failed lookups (f_{lookup}). Therefore, we propose to use this replication mechanism in settings with up to 4 clusters.

Finally, we evaluated our fifth replication mechanism. As DRA has been developed for settings with an increased number of clusters, we compared this mechanism with OCRA in settings with 4 and 16 clusters as required by our second evaluation goal. We were able to show that DRA distributes replicas more efficiently as OCRA in networks with 16 clusters. Thus, we propose to use DRA whenever a higher amount of clusters is required.

As shown, replicas can be used to reduce the affects introduced by network partitioning in settings with a low node density. By distributing the replicas in the deployment area, the probability has been increased that at least one of the replicas is available when a root node or a set of nodes are disconnected from the network. Therefore, we were able to confirm our assumption as requested by our last evaluation goal.

In conclusion, replication as proposed by our mechanisms is highly recommended for MP2P systems as Clustered Pastry. Replicas increase strongly the robustness against maliciously behaving root nodes that perform the *Storage and Retrieval Attack* but also improve the availability of objects in sparse networks. We have further shown that adapting the replication mechanism to the structure of the specific MP2P system can be beneficial. Yet, the basic structure of our replication mechanisms can be easily mapped on the structure of other clustered or location aware MP2P systems like MADPastry [118].

6.3 SECURE MESSAGE FORWARDING

Besides maliciously behaving root nodes, intermediate nodes may behave maliciously as well. Intermediate nodes are required in order to forward lookup messages towards the destination node. As a result, those nodes are able to affect the lookup functionality of an MP2P system, as discussed in Chapter 5.2.2. When a network is under the *Incorrect Lookup Routing Attack*, lookup messages are not forwarded correctly by intermediate nodes but are dropped or misrouted. Thus, the availability of the network services is affected.

Multiple mechanisms have been introduced to provide robustness against those attacks for P2P systems. However, as mentioned in Chapter 2.2.3, those approaches are mostly based on introducing redundancy to the lookup mechanism. Yet, this does not seem promising in the context of a wireless mobile environment with strongly limited resources in terms of bandwidth. Therefore, we discuss existing security mechanisms in the light of MP2P systems in this section. Moreover, a new approach is introduced that harnesses the characteristics of the wireless channel to detect malicious behavior. Thus, by overhearing messages sent by the geographical neighbors, this mechanism is able to monitor the route of a lookup request and, therefore, to detect dropped or rerouted requests.

6.3.1 *Challenges in Mobile Peer-to-Peer Systems*

The *Incorrect Lookup Routing Attack* exploits the characteristics of the DHT in order to deny the lookup services provided by the network. Therefore, we have to consider the challenges introduced by the overlay, in particular the decentralized routing mechanism and the required trust on intermediate nodes during a lookup. As the lookup is not coordinated by a central instance, malicious behavior is hard to detect. This is even more challenging as the recursive lookup mechanism, which is used by Clustered Pastry, is based on intermediate nodes that forward lookup request messages.

However, further challenges arise due to the MANET underlay. On one hand, we have to consider limited bandwidth when designing a security mechanism. This is very challenging especially as most existing mechanisms that have been developed for DHTs based on the Internet as underlay are based on introducing redundancy. On the other hand, an increased fraction of lost messages is introduced by the wireless channels, e.g., due to collisions during the transmission of two messages.

6.3.2 *Existing Security Mechanisms in Mobile Peer-to-Peer Networks*

As already discussed in Chapter 2.2.3, several security mechanisms have been developed in the last decade to improve the robustness of a DHT to the *Incorrect Lookup Routing Attack*. Most of these mechanisms either introduce redundancy to the lookup process or provide feedback to a coordinating instance during routing and are, therefore, based on the *Redundant Routing* [17] or the *Iterative Routing* [99] approach. Thus, we will focus on those two mechanisms in the following paragraphs.

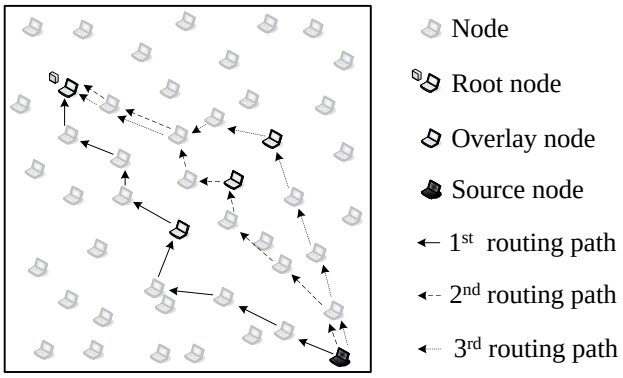


Figure 47: An example of a lookup with the *Redundant Routing* mechanism

REDUNDANT ROUTING

The Redundant Routing mechanism has been proposed by Castro et al. and is based on introducing redundancy to the lookup mechanism. Therefore, multiple request messages are sent in parallel whenever a lookup is initiated as shown in Figure 47. As a result, a lookup only fails when all of these request messages are dropped. However, in order to operate reliably, disjoint routing paths are required. Otherwise, a single malicious node is able to drop all of the request messages.

The robustness introduced by this mechanism can be displayed in terms of the fraction of failed lookups ($\sigma_{Redundant}$). This metric is a function of the fraction of malicious nodes (f), the average number of required overlay hops per lookup (h), and the number of requests that have been sent in parallel (s) as described by Equation 6.11

$$\sigma_{Redundant} = 1 - (1 - (1 - f)^{h-1})^s \tag{6.11}$$

As each of the request messages is routed recursively, the average number of overlay hops ($h_{Redundant}$) is equal to the one of a recursive routing mechanism ($h_{Recursive}$). Thus, the average number of overlay hops ($h_{Redundant}$) is a function of the fraction of malicious nodes in the network (f) and the number of average hops of a successful request (h) as described by Equation 6.12.

$$h_{ILR} = \sum_{i=0}^{h-1} (1 - f)^{i-1} \tag{6.12}$$

Also the average delay introduced by a lookup is not affected by the *Redundant Routing* mechanism. Yet, the average number of sent messages per lookup ($m_{Redundant}$) is increased by the factor s due to request messages that are sent in parallel as described by Equation 6.13.

$$m_{Redundant} = h_{Recursive} * s \tag{6.13}$$

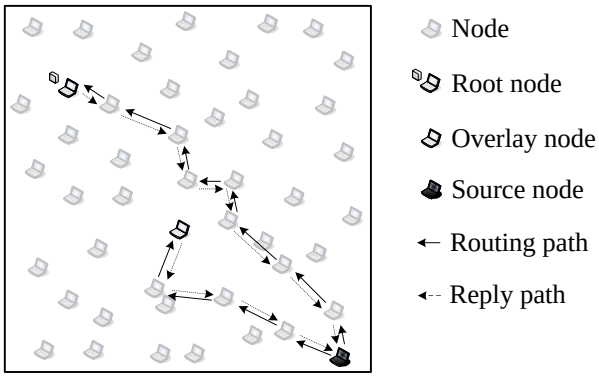


Figure 48: An example of a lookup with the *Iterative Routing* mechanism

ITERATIVE ROUTING

The *Iterative Routing* [99] mechanism is based on a coordinating instance. This instance receives feedback during the lookup process and, therefore, is able to detect whenever a node behaves maliciously. Sit and Morris proposed to harness the source of the lookup as coordinating instance. Whenever a node receives a request message, this request message is not forwarded but a reply message has to be sent to the source node is shown in Figure 48. This reply message includes a set of next hop addresses. As a result, the source node is able to coordinate the lookup and whenever no reply message is received, the request can be forwarded to any other node that has been listed in the last received reply message.

The fraction of failed lookups introduced by an *Iterative Routing* mechanism ($\sigma_{Iterative}$) is a function of the fraction of malicious nodes (f) and the average number of overlay hops (h). Moreover, the number of next hop addresses provided by each reply message (r) affects the robustness provided by this mechanism as described by Equation 6.14.

$$\sigma_{Iterative} = (1 - f^r)^{h-1} \tag{6.14}$$

Due to the feedback that is required to coordinate the lookup, the average number of messages sent per lookup ($m_{Iterative}$) and the average number of overlay hops per lookup ($h_{Iterative}$) is at least twice as high compared to a recursive approach. The precise number of hops and messages is described by Equation 6.15. Moreover, the delay is increased as the request messages are not directly forwarded by the intermediate nodes, but are sent to the source node first.

$$m_{Iterative} = h_{Iterative} = \sum_{i=0}^{h-2} (\sum_{j=0}^{r} f^j (1-f))^i * (\sum_{k=0}^{r} f^k + f^k * (1-f)) \tag{6.15}$$

DISCUSSION

Both mechanisms that have been discussed previously can be used to increase the networks robustness to the *Incorrect Lookup Routing Attack*. However, both of these mechanisms introduce overhead in terms of traffic. The *Redundant Routing* mechanism generates traffic due to the request messages that are sent in parallel, while the *Iterative Routing* mechanism introduces traffic as a result of the feedback functionality.

However, as the bandwidth is limited by the mobile, wireless underlay, overhead introduced by the routing mechanism should be minimized (as discussed in the previous section).

Moreover, both introduced mechanisms have a major drawback. The *Redundant Routing* requires disjoint routing paths in order to sent multiple request messages in parallel. Yet, this requirement is hard to satisfy as all parallel lookups are forwarded to the very same destination. The iterative approach on the other hand, is based on providing feedback to the source node. Yet, this approach is contrary to our Clustered Pastry approach, where the lookup distance is reduced with each hop in order to reduce overhead and the probability of collisions.

6.3.3 *Overlay WatchDog*

As shown in the previous paragraphs, existing security mechanisms that provide robustness to the *Incorrect Lookup Routing Attack* are based on introducing redundancy and, therefore, traffic overhead. Thus, they are not efficient in the context of a wireless MP2P system with a strongly limited bandwidth.

Therefore, we propose a new security mechanism that has been developed in the context of a MP2P system. Thus, this mechanism is able to meet the challenges discussed previously. Our approach benefits from the characteristics of wireless transmission to detect dropped and misrouted request messages. In the following section, we discuss our new *Overlay WatchDog* mechanism. Therefore, we introduce the requirements that arise from this mechanism. Thereafter, we define the detection and response algorithm of our *Overlay WatchDog*.

ASSUMPTIONS AND REQUIREMENTS

Our *Overlay WatchDog* mechanism is based on a set of requirements that have to be met in order to ensure a reliable operation of the security mechanism. These requirements are surveyed in the following list.

- Each node that in part of the MANET has to participate in the MP2P network. This assumption is satisfied in disaster relief scenarios, where each device is preinstalled and, therefore, uses the same software.
- Each node has to be able to overhear messages sent by geographical neighbors. Thus, the participating nodes have to operate in the promiscuous mode. However, passive monitoring systems can be used to sniff packets in an efficient way [96].
- Each node has to monitor the quality of links to geographical neighbors. Based on this information, a node is aware of nodes that leave the transmission range. However, several MANET routing algorithms provide this information including the OLSR ETX protocol that is used by our Clustered Pastry system.
- Each node has to be able to identify lookup request messages at the underlay. Moreover, the IP address of the next underlay hop and the overlay identifier of the next overlay hop have to be determined at the underlay. While the IP address of the next underlay hop is provided by the routing table of the MANET, the overlay identifier of the next overlay hop has to be extracted from the request message.

DETECTION OF MALICIOUS BEHAVIOR

Due to the structure of the DHT overlay, multiple overlay hops are required to resolve a lookup as discussed in Chapter 2.1.3. Each of these overlay hops corresponds to a complete underlay route.

An example for a lookup is shown in Figure 49. Node A is the source of the lookup request. Node B is an intermediate overlay hop and node C is the destination of this lookup. The nodes a, b, c, and d are intermediate underlay hops. When node A initiates a lookup for node C, the routing table of A provides the IP-address and overlay identifier of node B. Therefore, a request message is sent to node B. However, as node B is not within a transmission range of node A, a route via node a and b is established by the MANET underlay. Yet, neither node a nor node b participate in the overlay routing but only forward the message to the next underlay hop until the underlay destination is reached. After receiving the lookup request, node B determines the next overlay hop and forwards the lookup. Once again, the next overlay hop is not within transmission range and intermediate underlay hops are required. After forwarding the request via node c and d to the next overlay hop C, the lookup is received by the destination node. However, when considering the *Incorrect Lookup Routing Attack*, the lookup request can be dropped by an intermediate overlay node. However, as node A is the source and node C the destination of the lookup in our example, only node B may behave maliciously and drop the lookup.

Marti et al. introduced the WatchDog IDS in order to detect maliciously behaving nodes in a MANET. Whenever a message is forwarded in the MANET, this IDS is used to monitor the behavior of the next hop node. If this next hop node does not forward the message and is not the destination of this message, a malicious behavior is detected by the WatchDog IDS. For example considering the route between node A and node B. When node A forwards a message to node a, node A is aware that a is not the destination of this message but only an intermediate underlay node. Therefore, node A monitors the messages send by node a until the message is forwarded to node b. Thereafter, node a has to monitor messages sent by node b. As a result, malicious behavior can be detected in the underlay without introducing traffic overhead.

However, WatchDog can only be used to detect maliciously behaving intermediate underlay nodes as only the underlay route can be monitored by WatchDog. For example, when node b forwards the request message to node B, this node is identified as the destination of the message due to the IP-address of node B. Yet, when considering the lookup process, node B is not the destination of the lookup but only an intermediate overlay hop.

Therefore, we propose the *Overlay WatchDog* that is based on WatchDog but also considers overlay information to detect maliciously behaving overlay nodes. Each node has to be able to identify request messages at the underlay. Whenever a request message is detected, the *Overlay WatchDog* checks whether the next hop node is the destination of this underlay route. If so, the next hop node must be involved into the overlay routing and, therefore, has either to forward the request to a node that is logically closer to the destination or send a reply message. This can be monitored by the last underlay hop. However, in order to be able to detect misrouted request messages, this intermediate underlay node has to extract the identifier of the destination of the lookup from the lookup request and compare this identifier with the identifier of the overlay node and the next overlay hop. For example, let us consider node b forwards the request to node B. As node B is the destination of the

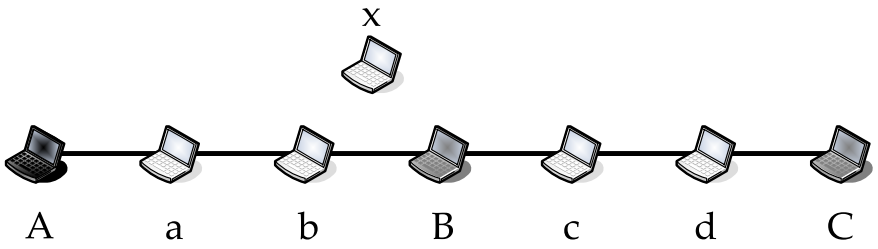


Figure 49: An example of an Mobile Peer-to-Peer lookup

underlay route, node B has to be involved in the overlay routing. As a intermediate overlay node, node B has to forward the lookup request to node C. This message can be detected by node b. However, when node B drops or misroutes the request, node b is able to detect this misbehavior.

However, we also have to consider colluding maliciously behaving nodes. Therefore, we have to address a case where an intermediate underlay node does not report a malicious behavior of the next hop overlay node. Yet, all nodes that are in transmission range of the colluding nodes are able to detect the malicious behavior as well. Thus, we can use those nodes as well in settings with colluding maliciously behaving nodes. Let us assume that both, node b and node B behave maliciously. As a result, node b cannot be used to detect messages that have been dropped by node B. Yet, node x is within transmission range of node b and node B and, therefore, is able to detect that a request message has been forwarded by node b but dropped at node B.

REACTION TO MALICIOUS BEHAVIOR

When a node detects a dropped or misrouted lookup message, action must be taken to ensure that the lookup does not fail due to this malicious behavior. Initially, the node that responds to this malicious behavior has to be chosen. Thereafter, we have to define how this node responds to a dropped or redirected request message.

Each message that is dropped or misrouted by maliciously behaving intermediate overlay node can be detected by the underlay node, which has forwarded this message to the malicious node. Thus, we prefer this node to respond to this malicious behavior. Yet, as we assume malicious colluding nodes, we have to consider a malicious behavior of this previous underlay hop node as well. Therefore, also nodes have to be considered that are within transmission range of the maliciously behaving node and the previous underlay hop node and, therefore, are also able to detect dropped or misrouted messages. Thus, whenever the previous underlay hop node does not respond to a detected malicious behavior, these other nodes have to respond.

After the responding node has been selected, this node has to forward the lookup request. As this node is geographical close to the maliciously behaving node, it is most probable a virtual neighbor of the malicious intermediate node as well. Due to this fact, both nodes provide a similar set of routing table entries. As a result, the responding node is able to determine the next overlay hop node and to forward the lookup request efficiently to this node without introducing a high traffic overhead. Furthermore, the node that has been dropped the request message is avoided in the subsequent routing steps.

DISCUSSION

Our *Overlay WatchDog* security mechanism uses overheard request messages that are sent by the geographical neighbors to detect dropped or misrouted messages. Thus, only a low traffic overhead is introduced by this approach. Whenever a message is dropped or misrouted, a new request message is sent by the node that has detected this malicious behavior.

As described by Equation 6.16, the mechanism provides robustness to the *Incorrect Lookup Routing Attack* and, therefore, the fraction of failed lookups (σ_{OW}) is strongly reduced. However, this fraction is a function of the average number of overlay hops (h), the fraction of malicious nodes (f), and the number of stored addresses per routing table entry (r).

$$\sigma_{OW} = (1-f) * (\sum_{i=1}^{r} (\sum_{j=1}^{n} f^j * (1-f))^i * (1-f) + (1-f))^{h-1} \tag{6.16}$$

The average number of overlay hops of the Overlay WatchDog (h_{OW}) is equal to the average number of overlay hops of a recursive routing mechanism in a setting without maliciously behaving nodes. The same applies for the average number of sent messages (m_{OW}) and the delay. Yet, when increasing the fraction of malicious nodes (f), all of these three metrics are affected. As described by Equation 6.17, the number average hops is slightly increased when introducing maliciously behaving nodes.

$$\begin{aligned} m_{OW} = h_{OW} = \sum_{i=0}^{h-2} (\sum_{k=1}^{r} (\sum_{l=1}^{n} f^l)^k * (1-f)^k + 1 - f)^i * \\ (\sum_{k=1}^{r} (\sum_{j=1}^{n} f^j * (1-f))^k * (1-f) + (1-f))^{h-1} \end{aligned} \tag{6.17}$$

6.3.4 *Conclusions on Secure Message Forwarding*

In this section, we discussed the efficiency of security mechanisms that provide robustness to the *Incorrect Lookup Routing Attack* in MP2P scenarios. Therefore, we analyzed the *Iterative Routing* and the *Redundant Routing* mechanism under the light of a limited bandwidth and a wireless transmission channel. As a result, we were able to identify drawbacks introduced by these approaches that limit the usability of these security mechanisms in settings based on a MP2P system like Clustered Pastry. Thus, we developed a new security mechanism for MP2P systems, which benefits from the characteristics of the wireless data transmission. This *Overlay WatchDog* uses overheard messages to detect dropped or misrouted messages and, therefore, introduces an overall low overhead in terms of traffic.

6.4 EVALUATION OF THE OVERLAY WATCHDOG MECHANISM

In the previous section, we have discussed three different security mechanisms that can be used to introduce robustness to the *Incorrect Lookup Routing Attack*. The *Redundant Routing* and the *Iterative Routing* mechanism have been proposed for P2P

systems that are based on a static underlay as the Internet. The third approach has been developed in the context of this thesis and is based on the characteristics of an MP2P system. However, like most DHTs Pastry and, therefore, Clustered Pastry is not able to provide the disjoint routing paths that are required by the *Redundant Routing* mechanism. Therefore, and as the other mechanisms provided better results in the analytic evaluation, we do not consider the *Redundant Routing* in the context of this simulative evaluation

In the following section, we evaluate the *Iterative Routing* mechanism and our *Overlay WatchDog* in MP2P settings.

6.4.1 *Evaluation Goals, Metrics, and Methods*

In the next paragraphs we define our basic evaluation goal and introduce the used metrics for the following evaluation. Moreover, we discuss our evaluation methods.

EVALUATION GOALS

Two evaluation goals have to be satisfied in the following evaluation in order to determine the efficiency of the security mechanisms. On one hand, those security mechanism have to be able to operate in an MP2P network efficiently. Therefore, we have to analyze the impact of these mechanisms on MP2P systems with different scale. On the other hand, the robustness against the *Incorrect Lookup Routing Attack* that is provided by the security mechanism has to be determined.

EVALUATION METRICS

The following evaluation is based on two metrics. The first metric is the fraction of failed lookups (f_{lookup}) and is used to introduce the efficiency of our security mechanism and the impact of the *Incorrect Lookup Routing Attack*. Thus, we further use this metric to analyze the robustness provided by the security mechanism. Moreover, we harness the overall traffic (T) generated by the MP2P system in order to determine the overhead and, therefore, the costs introduced by the security mechanism.

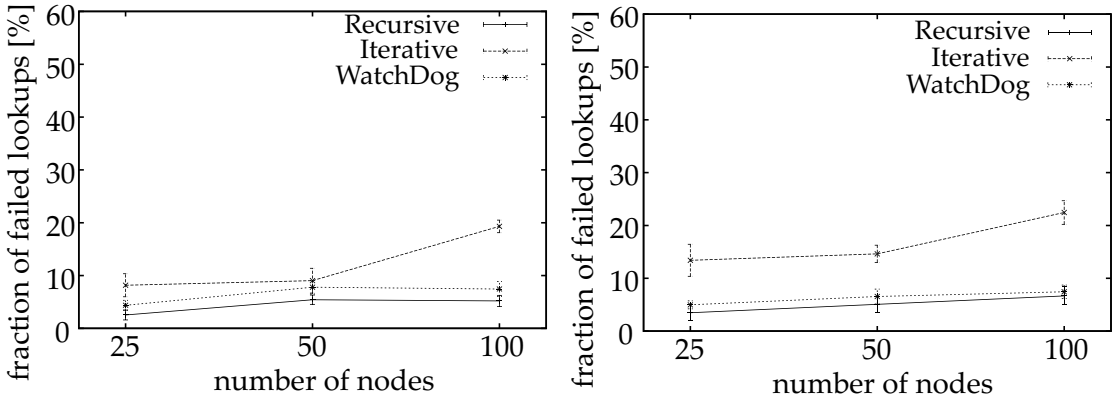
EVALUATION METHODS

We used our Clustered Pastry model, which has been implemented for the OMNeT++ simulator, to evaluate our *Overlay WatchDog*. Therefore, we implemented our new security mechanism as discussed in the previous section. Moreover, we deployed the *Iterative Routing* mechanism provided by the OverSim framework as reference model to our *Overlay WatchDog* mechanism. During the evaluation, the basic parameters as defined in Chapter 4.1.4 are used.

6.4.2 *Comparison of Security Mechanisms*

In this first part of the evaluation we analyze the efficiency of the *Iterative Routing* mechanisms and the *Overlay WatchDog* in the light of MP2P systems. Thus, we simulate and discuss settings without maliciously behaving nodes but with a different network size in the following paragraphs.

We simulate settings with 25 to 100 nodes with 2 and 4 clusters in order to show the influence of those scenarios on both security mechanisms. Moreover, the outcome



(a) 2 Clusters (b) 4 Clusters

Figure 50: Usability of the *Iterative Routing* and *Overlay WatchDog* mechanism in Clustered Pastry

of these simulations are compared to the results provided by a setting without any security mechanisms that is based on the recursive routing mechanism only. We evaluate the efficiency of these mechanisms based on the fraction of failed lookups (f_{lookup}).

RESULTS OF THE COMPARISON OF SECURITY MECHANISMS

The recursive routing mechanism provides reliable results in settings without maliciously behaving nodes as discussed in Chapter 4.5. Thus, more than 95% of all lookups can be performed successfully. However, as our *Overlay WatchDog* is only triggered by dropped or rerouted lookup messages, a very similar result is provided in settings with 2 or 4 clusters as shown in Figure 50a and 50b.

The *Iterative Routing* mechanism on the other hand is strongly affected by the number of clusters and the field size. This security mechanism uses the source node to coordinate lookups. Thus, request messages are not forwarded by intermediate nodes but reply messages are sent to the source node during each lookup step. As a result, the number of required lookup messages is doubled and, furthermore, the distance between the sender and the next overlay hop is increased on average with each hop. This increases the probability of failed lookup messages due to the wireless underlay (e.g., as a result of collisions). Due to this fact, the fraction of failed lookups is increased when we increase the network size as this affects the average distance between the source and destination. Moreover, by increasing the number of clusters, the reliability is reduced as well, as the clustering level defines the average number of hops.

SUMMARY OF THE COMPARISON OF THE SECURITY MECHANISMS

In the previous evaluation we were able to show that our *Overlay WatchDog* does not affect Clustered Pastry's lookup mechanism in settings without maliciously behaving nodes. Yet, the *Iterative Routing* approach is strongly affected by both the size of the setting in terms of the field size and the number of participating nodes, and the number of clusters. Thus, the fraction of failed lookups increases when either the average geographical distance between the source and destination node or the average

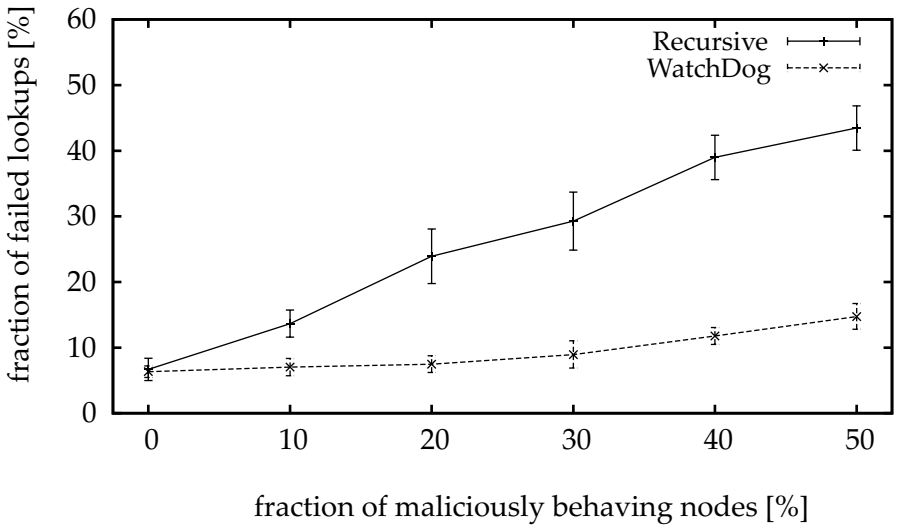


Figure 51: Robustness provided by the *Overlay WatchDog* to the *Incorrect Lookup Routing Attack*

number of required overlay hops is increased. Therefore, we were able to show that an *Iterative Routing* mechanism is not able to provide efficient results in a location aware MP2P system as Clustered Pastry.

6.4.3 *Robustness to the Incorrect Lookup Routing Attack*

Maliciously behaving nodes that drop or reroute lookup messages are able to affect the reliability of the services provided by the MP2P system. Therefore, a security mechanism is required to increase the networks robustness to this *Incorrect Lookup Routing Attack*.

In the following paragraphs, we analyze the *Overlay WatchDog* in settings with maliciously behaving nodes. Therefore, the fraction of failed lookups (f_{lookup}) and the overall traffic (T) are used as evaluation metric.

RESULTS OF THE ROBUSTNESS EVALUATION AGAINST THE INCORRECT LOOKUP ROUTING ATTACK

The *Incorrect Lookup Routing Attack* has an high impact on the reliability of the lookup mechanism as already discussed in the Chapter 5.2.2. However, the *Overlay WatchDog* is able to strongly reduce the fraction of failed lookups that is introduced by the maliciously behaving nodes as shown in Figure 51. For example, when considering a setting with 40% of maliciously behaving nodes, the fraction of failed loss can be reduced from a loss rate of 39% to less than 12% by using the *Overlay WatchDog* mechanism.

Moreover, only a small amount of overhead is introduced by our *Overlay WatchDog* as shown in Figure 52. In setting without maliciously behaving nodes, the traffic (T) is increased by less than 6% due to messages that have been dropped as a result of the wireless underlay and have been retransmitted by our *Overlay WatchDog*. However, the traffic of the recursive unsecured mechanisms decreases slightly when the fraction of malicious nodes is increased due to the increased number of failed

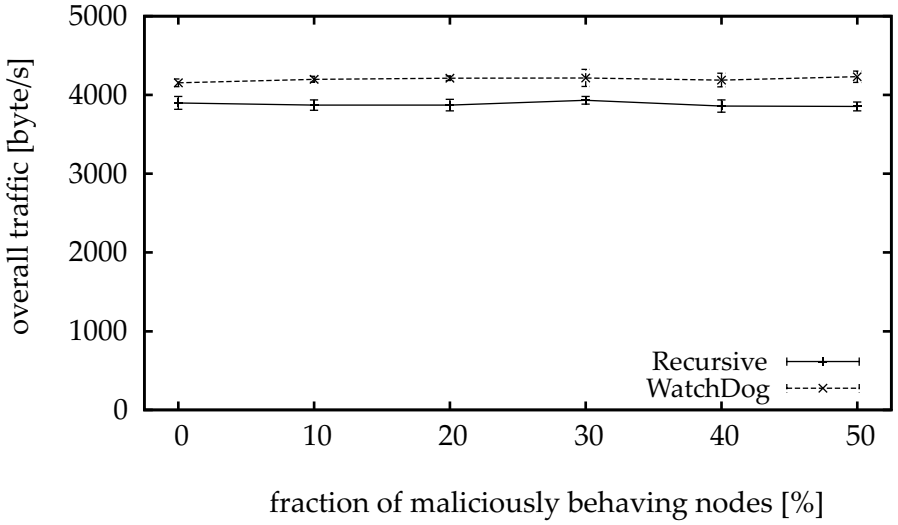


Figure 52: Traffic introduced by the *Overlay WatchDog* compared to the traffic of an unsecured network

lookups. Yet, our *Overlay WatchDog* introduces even an increased amount of traffic in settings with a high fraction of maliciously nodes due to the resulting increased number of retransmissions. Though, even in settings where every second node behaves maliciously the traffic is not increased by more than 9.2%.

SUMMARY OF THE ROBUSTNESS AGAINST THE INCORRECT LOOKUP ROUTING ATTACK

As shown in this subsection, our *Overlay WatchDog* is able to strongly decrease the impact of the *Incorrect Lookup Routing Attack*. Moreover, our *Overlay WatchDog* increases the overall traffic only by 6% when considering a setting without maliciously behaving nodes. As a result, we were able to show that our security mechanism provides robustness to the *Incorrect Lookup Routing Attack* and introduces a traffic overhead of less than 10%.

6.4.4 *Summary of the Overlay WatchDog Evaluation*

In this section, we discussed the shortcomings of existing security mechanisms that provide robustness to the *Incorrect Lookup Routing Attack*. Therefore, two mechanism, most other security mechanisms are based on, were surveyed and reviewed in the light of our Clustered Pastry system. The first approach requires disjoint paths in order to ensure reliable services. Though, disjoint routing paths are not provided by our Clustered Pastry system in settings without replicas. The second approach harness the source of the lookup to coordinate the route of the lookup request. Yet, this source based routing introduces overhead as lookup messages have to be replied to the source after each hop. This results in an increased fraction of failed lookups, especially in settings with an increased network size, as shown in our evaluation. However, we further evaluated our *Overlay WatchDog*. Our new approach was able to provide robustness to the *Incorrect Lookup Routing Attacks*, as shown in our evaluation. Moreover, only a small traffic overhead has been introduced by our mechanism. As

a result, we propose to use our *Overlay WatchDog* in the context with our Clustered Pastry system in order to provide robustness to the *Incorrect Lookup Routing Attack*.

6.5 VALIDATING THE ROOT NODE

Whenever a malicious node receives a lookup request, this node may either drop or redirect this request or forge a reply message. The *Overlay WatchDog* mechanism that has been introduced in Chapter 6.3.3 is able to detect dropped and redirected messages but fails to detect forged reply messages. Therefore, a mechanism is required in order to validate received reply messages.

6.5.1 *Challenges in Mobile Peer-to-Peer Systems*

Due to the wireless underlay and the characteristics of the disaster relief scenarios, four challenges for validation mechanisms arise. The first three challenges are introduced by most MP2P systems. However, the fourth challenge has only to be considered in context with location aware MP2P systems as Clustered Pastry or MADPastry [118]. These challenges are as follows:

I No highly available or static entity may be assumed in MP2P systems. Therefore, no static entities that keep track on the virtual identifiers of the nodes in the network are available in the scenario considered in this thesis.

II The mobile underlay of an MP2P system introduces a strongly limited bandwidth. Therefore, the traffic introduced by security mechanisms has to be minimized.

III Due to limited bandwidth, the routing tables of most MP2P systems are truncated in order to reduce the traffic introduced by the update mechanism of the routing tables. As a result, only a strongly limited number of virtual neighbors and distant nodes is stored at the systems routing tables.

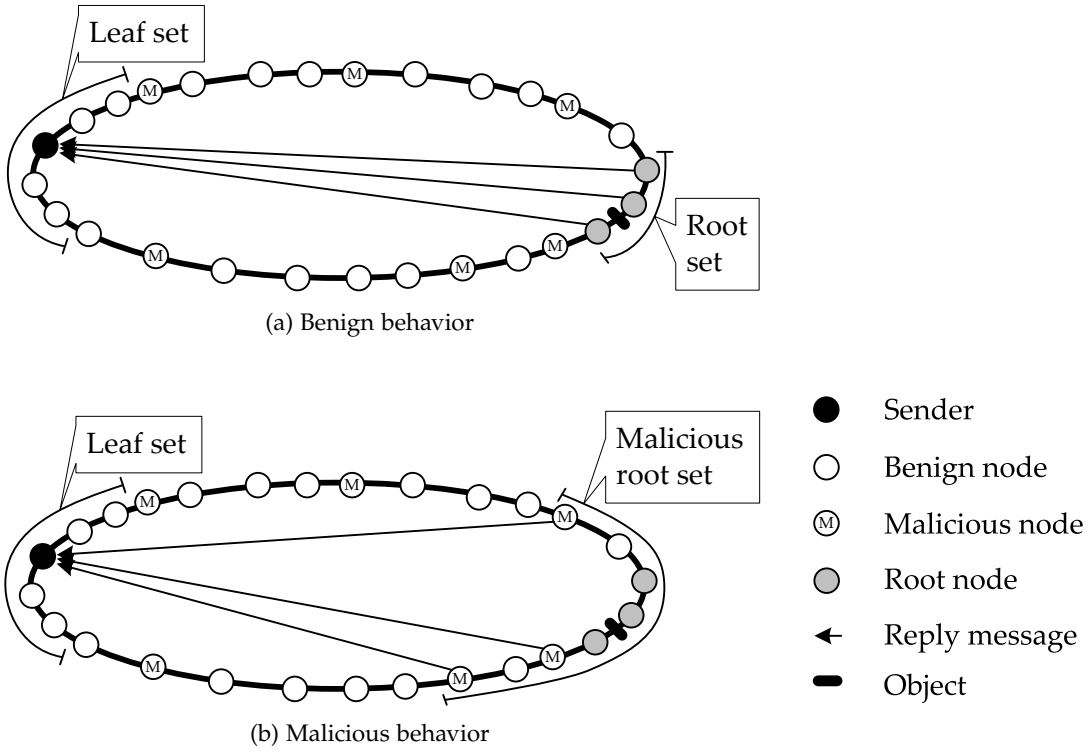
IV Location aware MP2P systems determine the node identifiers from the geographical position of the node. As a result, overlay identifiers of nodes are not uniformly distributed in the identifier namespace.

6.5.2 *Existing Validation Mechanisms in Mobile Peer-to-Peer Scenarios*

In Chapter 2.2.4, three approaches to validate a reply messages have been discussed, which have been proposed by the related work in the context of DHTs. The security mechanisms introduced by Wang et al. [114] and Ganesh and Zaho [30] are both based on a highly available instance (I) and, therefore, cannot be used in the context of our disaster relief scenario. The third approach has been proposed by Castro et al. [17] and does not require central or highly available entities but is based on assumptions that cannot be satisfied in the discussed disaster relief scenario. However, we will discuss this approach in depth in the following paragraphs.

ROUTING FAILURE TEST

Castro et al. introduced the *Routing Failure Test*, which is based on the following



(a) Benign behavior

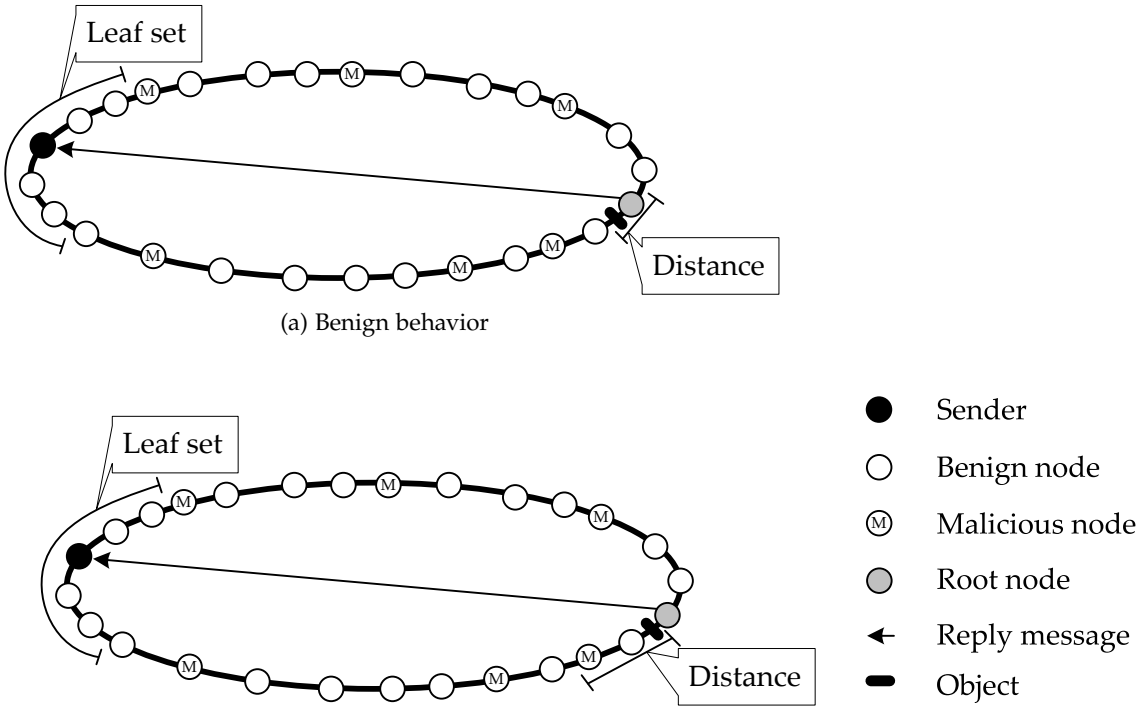
(b) Malicious behavior

Figure 53: Example for the derived node density in a *Routing Failure Test*

assumptions. The overlay identifiers of the nodes have been generated randomly and are, therefore, uniformly distributed in the identifier space. Each node has to maintain links to a set of virtual neighbors. Moreover, replicas have to be stored at the virtual neighbors of the root.

Per default, a reply message only contains the address of the root node. Castro proposed to expand the reply message by the addresses of the virtual neighbors of this root node, as those neighbors maintain the replicas of the requested object as stated above. This resulting set of nodes is called the root set. An example for a root set of a requested object is shown in Figure 53a. As long as malicious nodes do not collude, malicious behavior can be detected with little effort. Whenever a malicious behavior is assumed by the sender of the request, this node has to contact another node of the root set. When this root set node is responsible for a replica of the requested object, the result is validated. In any other case, a malicious behavior can be assumed.

However, malicious nodes may collude in order to provide a root set that consists of maliciously behaving nodes only, as shown in Figure 53b. To detect such a malicious root set, Castro proposes to compare the average virtual distance of the root set with the average virtual distance of the sender's virtual neighbors. When assuming uniformly distributed overlay identifiers of the nodes, the average virtual distance between nodes in those two identifier sets should be equal. Otherwise, a malicious behavior is assumed. Castro evaluated this approach analytically in a setting with 100,000 nodes, 32 replicas, and a set of 32 virtual neighbors stored in the routing table of each node. As a result, the *Routing Failure Test* is able to detect forged reply message with a probability of more than 99%.



(a) Benign behavior

(b) Malicious behavior

Figure 54: Example for the derived node density in an *Adapted Routing Failure Test*

Yet, the requirements for the *Routing Failure Test* are contrary to the previously discussed challenges and limitations of MP2P systems. Thus, we cannot assume that the node identifiers are uniformly distributed in the namespace (IV). Moreover, a high number of overall nodes, replicas (III), or routing table entires that provide links to a large set of virtual neighbors (II) cannot be considered in an MP2P system. Thus, the *Routing Failure Test* can most probable not be used efficiently in the context of a MP2P system.

6.5.3 *Adapted Routing Failure Test*

Resources as bandwidth are strongly limited in MP2P systems. Therefore, only a limited number of replicas can be maintained in such a wireless, mobile network. Otherwise the efficiency of the lookup mechanism would suffer due to congestion generated by uploading and updating the replicas.

Nevertheless, harnessing the node density to detect forged reply messages seems to be a promising approach at first glance. When no replica set is available or when the replicas are not stored at the virtual neighbors, the virtual distance between the object identifier and the identifier of the root node can be compared with the average distance of the virtual neighbors of the requesting node. An example for correct and a forged reply message is shown in Figure 54a and Figure 54b. The virtual distance between the root node and the object is rather short as compared to the virtual distance between the closest malicious node and the object. Thus, malicious behavior can also be identified in a setting without replicas.

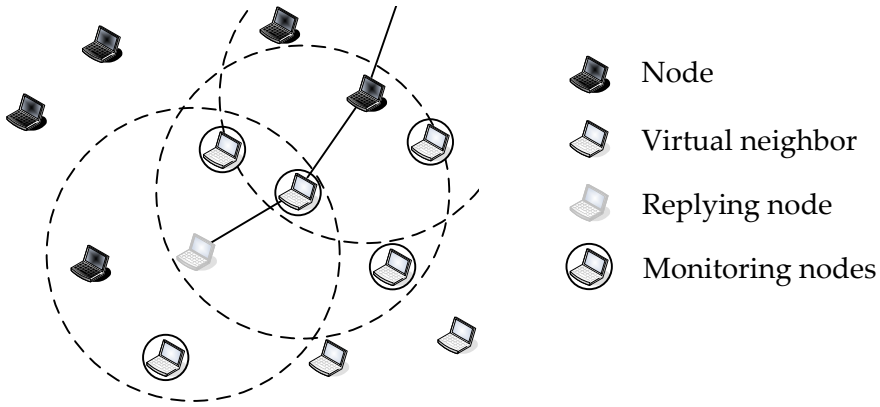


Figure 55: Example for the *Cross-Layer* approach

However, this approach still does not satisfy all of the previously introduced challenges of MP2P systems. The *Adapted Routing Failure Test* still requires a large set of virtual neighbors (III) in order to derive a reliable average distance between the nodes. Furthermore, the overlay identifier of the nodes have to be uniformly distributed (VI).

6.5.4 *Cross-Layer Validation Mechanism*

As mentioned earlier, multiple MP2P systems including Clustered Pastry harness the geographical location of nodes in order to determine the node identifiers. As a result, virtual neighbors of a node are geographically arranged around this node. This improves the efficiency of both, the lookup mechanism and the update mechanism of the MP2P system. Furthermore, the detection of incorrect and forged reply messages can also benefit from the correlation of the geographical and virtual location of the MP2P nodes.

Due to the structure of the DHT-based overlay, each node is aware of the objects a virtual neighbor is responsible for. Thus, each virtual neighbor of a node that claims to be the root of an object is able to validate this claim. As the MANET underlay uses a wireless channel, messages may be overheard by the geographical neighbors of the senders of these messages. Due to this fact, a reply message can be received by at least a fraction of the virtual neighbors of the replying node in a Clustered Pastry network. Therefore, those neighbors are able to validate overheard reply messages. When a forged reply message is detected, a notification message has to be sent to the requesting node. This reply message may include a proposed next hop address in order to enable the requesting node to complete the lookup correctly.

An example for this *Cross-Layer Validation Mechanism* is shown in Figure 55. The replying node has seven virtual neighbors that are geographically arranged around this node due to the location aware distribution of the overlay identifiers. However, only those nodes that are within the transmission range of the replying node or that are located along the route back to the requesting node receive this reply message. As a result, only a subset of the virtual neighbors can be used for the detection of a forged reply messages. However, still five nodes would be able to notify the sender in the case of a malicious behavior in our example.

6.5.5 *Conclusions on the Validation of Reply Messages*

Forged reply messages can be used to deny the services provided by the MP2P system. Thus, mechanisms are required to validate received reply messages. As discussed in this section, existing mechanisms that have been proposed to validate reply messages in the context of DHT systems that are based on the Internet as underlay do not suffice the requirements of our scenario. Therefore, we developed the *Adapted Routing Failure Mechanism* that is able to handle most of the challenges introduced by MP2P systems. Moreover, we proposed the *Cross-Layer Validation Mechanism* that uses the location awareness of nodes provided by systems as our Clustered Pastry. Based on this validation mechanism, we are able to detect forged reply messages in settings where node identifiers are not generated randomly but are a function or, e.g., the geographical location of the node.

6.6 EVALUATION OF THE DESTINATION VALIDATION MECHANISMS

In the previous paragraphs, three different validation mechanisms have been discussed. Two of these approaches, the *Routing Failure Test* proposed by Castro et al. and our adapted version of this test, harness the density of the uniformly distributed overlay identifiers in the MP2P system to detect forged or faulty reply messages. The *Cross-Layer Validation Mechanism* uses overheard messages and the geographical clustering of the overlay identifiers in order to detect forged reply messages in location aware MP2P systems.

In this section, we evaluate all three mechanisms in the light of MP2P systems.

6.6.1 *Evaluation Goals, Metrics, and Methods*

In the following paragraphs, we define the evaluation goal and discuss the metrics, which are harnessed to evaluate the validation mechanisms. Moreover, the evaluation methods are introduced.

EVALUATION GOALS

Our major evaluation goal is to analyze and compare the efficiency of the previously discussed validation mechanisms in an MP2P scenario. Therefore, limitations and characteristics of MP2P systems have to be considered. Moreover, we have to consider the trade-off between the fraction of correctly detected malicious and falsely accused benign nodes.

EVALUATION METRICS

The detection ratio is harnessed as metric to determine the efficiency of the three security mechanisms. Therefore, on one hand, the true positives ratio (f_{TP}) is used to determine the efficiency of the validation mechanism. The true positives are the fraction of the correctly detected malicious behavior to the overall number of forged reply messages. Furthermore, the false positives ratio (f_{FP}) is used to indicate the fraction of valid reply messages that are detected as malicious by the validation mechanisms.

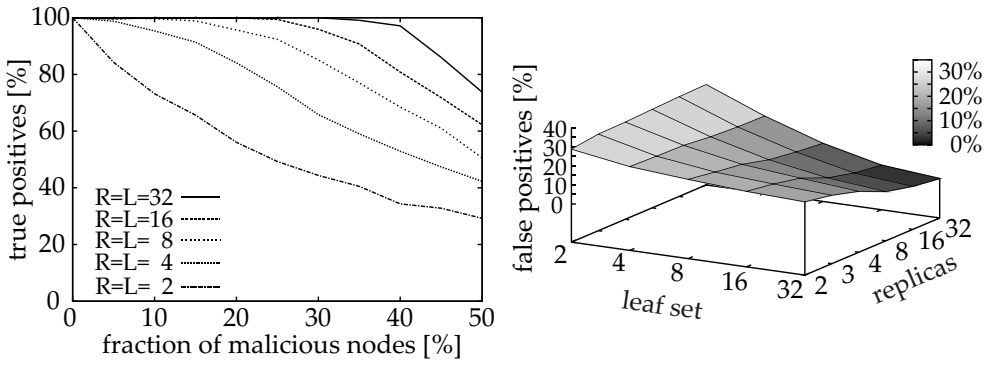
EVALUATION METHODS

We implemented the discussed mechanisms in order to evaluate their efficiency regarding the detection of forged reply messages by means of simulation. Yet, our Clustered Pastry system does not satisfy the assumptions of the *Routing Failure Test* and the *Adapted Routing Failure Test* regarding the distribution of node identifiers. Therefore, our Clustered Pastry model, as described in Section 3 cannot be used for the evaluation of these two approaches. Thus, we do not use the OMNeT++ simulator, but we implemented an abstract simulator to evaluate the two versions of the *Routing Failure Test*.

This simulator is based on the algorithms of the Pastry DHT. During each simulation run, a set of nodes with randomly distributed overlay identifiers is generated. Based on this set of nodes, the *leaf set* of each node is determined. In a second step, objects are generated and stored at the appropriate root node or the set of root nodes, respectively. Moreover, nodes may be marked as malicious. During each simulation run, two settings are analyzed. After randomly selecting a requesting node and an object, which has to be retrieved, a benign and a malicious setting are simulated. For the benign setting, the average virtual distance of the *leaf set* is compared to the average virtual distance of the root set or with the distance between the object and the root node, respectively. This setting is used to determine the false positives (f_{FP}). For the second setting the average virtual distance of the *leaf set* is compared to the average virtual distance of the malicious root set or with the distance between the object and the malicious node that is logically closest to the object, respectively. This second set is used to determine the true positive rate (f_{TP}). Moreover, we introduce a factor x that introduces a tolerance for the variance of the node identifiers density. Thus, a reply message is assumed as malicious when the average distance of the replied (malicious) root set or the distance between the (malicious) root node and the object exceeds the average distance of the *leaf set* multiplied with x. As a result, this factor can be used to reduce the fraction of falsely accused nodes (f_{FP}). However, this also affects the fraction of correctly detected malicious nodes (f_{TP}). Moreover, the results provided by this simulator matches the results of Castro et al.'s analytical evaluation.

During the evaluation of the *Routing Failure Test* an x of 1.7 is assumed as proposed by Castro et al. [17]. However, as we assume a reduced accuracy of the *Adapted Routing Failure Test*, the factor x is increased to 1.9 during the evaluation of this detection mechanism. However, a setting with 100 nodes is simulated. Furthermore, we vary the *leaf set* size (L) as the number of virtual neighbors, the number of replicas (R) and the fraction of malicious nodes during the evaluation of the *Routing Failure Test* and our *Adapted Routing Failure Test*.

In order to evaluate the *Cross-Layer Validation Mechanism*, the OMNeT++ implementation of the Clustered Pastry system as introduced in Chapter 3 is used. Moreover, we implemented a maliciously behaving nodes that forge reply messages whenever they receive a lookup request. Again, default parameters as defined in Chapter 4.1.4 are used.



(a) True positives

(b) False positives

Figure 56: Results of the evaluation of the original routing failure test

6.6.2 *Evaluation of the Routing Failure Test*

The *Routing Failure Test* as introduced by Castro et al. [17] has been proposed to validate received reply messages. This approach is based on equally distributed node overlay identifiers. Moreover, a set of replicas, that are stored at virtual neighbors of the root node and links to a large set of virtual neighbors are required by this validation mechanism.

In the following paragraphs, we analyze the *Routing Failure Test* in a set of different settings with a *leaf set* size and a number of replicas that is varied between 2 and 32. We use the fraction of falsely accused nodes (f_{FP}) and the fraction of correctly detected malicious nodes (f_{TP}) as metric to evaluate the efficiency of this mechanism.

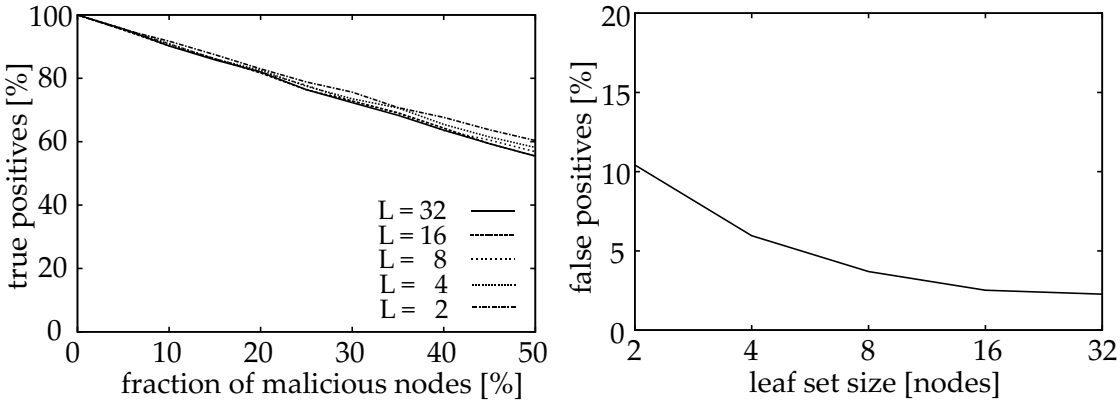
RESULTS OF THE ROUTING FAILURE TEST EVALUATION

The *Routing Failure Test* provides reliable results as long as the number of nodes in the *leaf set* (L) and the number of replicas (R) is high (e.g., $R = L = 32$). Malicious behavior can be detected correctly (f_{TP}) with a very high probability of more than 97% even when considering settings with up to 40% of maliciously behaving nodes, as shown in Figure 56a. However, when the number of *leaf set* nodes and stored replicas is reduced, the efficiency of the *Routing Failure Test* strongly degrades. For example, considering a setting with a *leaf set* size of 4 and 4 replicas, only 53% of the forged reply messages are detected by the validation mechanism when we consider a fraction of malicious nodes of 40%.

Also the fraction of benign nodes that are accused to behave maliciously is affected by the *leaf set* size and the number of replicas. As shown in Figure 56b, the false positive rate (f_{FP}) increases strongly when either of these two parameters is decreased. For example, in scenarios with 4 or less replicas, the false positive rate is increased beyond 10%.

SUMMARY OF THE ROUTING FAILURE TEST EVALUATION

The *Routing Failure Test* has been developed for settings with a high amount of nodes and a high bandwidth. Thus, a reliable detection of malicious nodes can be achieved in settings with 32 known virtual neighbors and high number of distributed replicas. These are required to avoid high variances in derived node density. As a result, the



(a) True positives

(b) False positives

Figure 57: Results of the evaluation of the adapted routing failure test

reliability of the *Routing Failure Test* degrades strongly when reducing either of those two parameters. However, most MP2P systems provide only a limited amount of replicas and links to virtual neighbors. Thus, the *Routing Failure Test* does not suffice the requirements introduced by MP2P systems.

6.6.3 *Evaluation of the Adapted Routing Failure Test*

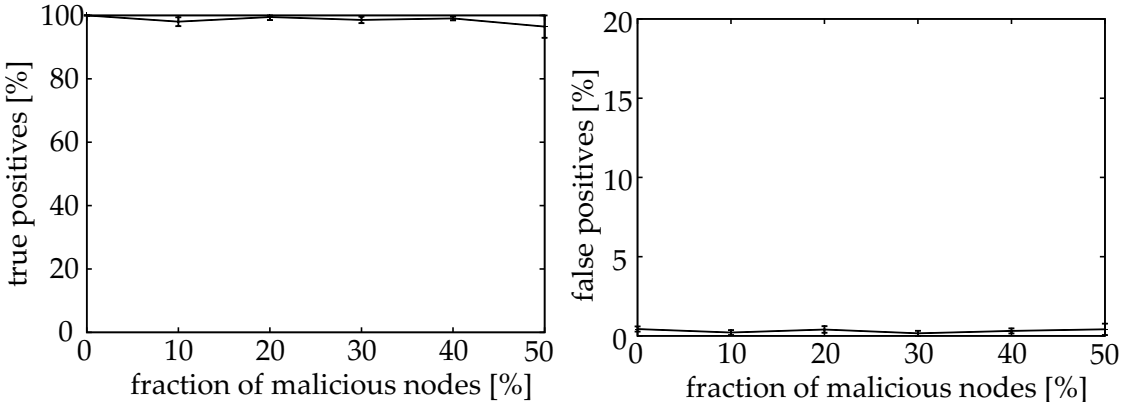
The *Adapted Routing Failure Test* is based on the *Routing Failure Test* and, therefore, harness the density of overlay identifiers to detect forged reply messages. Yet, this adapted approach does not require replicas but harness the virtual distance between the object identifier and the overlay identifier of the root node to detect forged reply messages.

We analyze the efficiency of this *Adapted Routing Failure Test* in settings with a *leaf set* size (L) between 2 and 32. Moreover we use the true positive (f_{TP}) and false positive rate (f_{FP}) to determine the efficiency of this approach.

RESULTS OF THE ADAPTED ROUTING FAILURE TEST EVALUATION

In contrary to the *Routing Failure Test* the *leaf set* size has nearly no influence on the true positive rate (f_{TP}) of the *Adapted Routing Failure Test* as shown in Figure 57a. Thus, better results can be achieved in settings with a small *leaf set* when we compare the results of *Adapted Routing Failure Test* with the *Routing Failure Test*. Though, still a high fraction of the forged reply messages is not detected correctly. For example, in a scenario with *leaf set* size of 4, only 65% of all forged reply message has been detected when considering a fraction of 40% of maliciously behaving nodes.

While the *leaf set* size has only a minor impact on the true positive rate (f_{TP}), the false positive rate is strongly affected by the number of virtual neighbors. Thus, the fraction of false positives is strongly increased in settings with a small *leaf set* as shown in Figure 57b. Even though the *Adapted Routing Failure Test* provides a lower false positive rate than the *Routing Failure Test* in settings with a low *leaf set* size, the false positive rate is quite high. For example, in a setting with *leaf set* size of 4, 6% of the correct reply messages are detected as forged messages.



(a) True positives

(b) False positives

Figure 58: Results of the evaluation of the cross layer validation of reply messages

SUMMARY OF THE ADAPTED ROUTING FAILURE TEST EVALUATION

The adapted approach is less efficient compared to Castro et al.'s *Routing Failure Test* when considering settings with a high number of replicas and a large *leaf set*. Yet, this approach performs better in settings with a small number of replicas or a small *leaf set* size as assumed in most of the MP2P scenarios. However, in settings with a high fraction of malicious nodes, the fraction of true positives degrades strongly.

6.6.4 *Evaluation of the Cross-Layer Validation Mechanism*

As both of the previously evaluated mechanisms require equally distributed node identifiers, these mechanisms cannot be used in the context of a location aware MP2P system as our Clustered Pastry. Therefore, we proposed a cross-layered mechanism that is able to detect forged reply messages based on overheard messages as discussed in Chapter 6.5.4.

In the following paragraphs, we evaluate this approach in combination with our Clustered Pastry system. Therefore, we use OMNeT++ to simulate settings with up to 50% of maliciously behaving nodes. As this approach requires virtual neighbors to detect maliciously behaving nodes, we use a setting with 50 nodes and 4 clusters to evaluate this validation mechanism as this result in a low number of virtual neighbors (12.5 virtual neighbors on average). As a result, we are able to ensure that the *Cross-Layer Validation Mechanism* operates efficient in a worst case scenario. Once again we use the true positive ratio (f_{TP}) and the false positive ratio (f_{FP}) to determine the efficacy of our approach.

RESULTS OF THE CROSS-LAYER VALIDATION MECHANISM EVALUATION

Our *Cross-Layer Validation Mechanism* harnesses the virtual neighbors of the replying node to detect forged reply messages. Thus, only a single benignly behaving virtual neighbor that is within transmission range of the reply route suffices to detect a forged message. During the simulation, we assumed that all maliciously behaving nodes collude and, therefore, do not notify the sender of the request when a malicious behavior is detected. However, our cross-layered validation mechanism is able to detect forged reply messages with a probability of more than 98% in a setting where

40% of all nodes behaves maliciously as show in Figure 58a. Thus, our *Cross-Layer Validation Mechanism* is able to provide better results as both versions of the *Routing Failure Test* in a comparable setting (no replicas, 12.5 virtual neighbors).

Moreover, the cross-layer approach provides a very low percentage of falsely accused nodes. Thus, the false positive ratio (f_{FP}) is below 0.6% as shown in Figure 58b. False positives are a result of stale routing tables and occur whenever a virtual neighbor has a faulty or outdated routing table entry that provides a links to a node that is virtually closer to the requested object than the root node.

SUMMARY OF THE CROSS-LAYER VALIDATION EVALUATION

Our cross-layered validation mechanism is able to detect forged reply messages with a very high probability even in settings with a high fraction of malicious nodes. Furthermore, the false positive ratio (f_{FP}) is very low. Yet, this approach can only be used by location aware MP2P systems as MADPastry or Clustered Pastry.

6.6.5 *Conclusions on the Validation of the Request Destination*

Maliciously behaving nodes are able to forge reply messages and, therefore, deny objects that are stored in the network. Therefore, a validation mechanism is required to detect forged reply messages and to ensure the availability of the networks services. In the last section, we evaluated the efficiency of three validation mechanisms in a MP2P scenario.

The *Routing Failure Test* has been proposed by Castro et al. [17] and compares the density of the virtual neighbors of the requesting node with the virtual distance between the set of root nodes in order to validate incoming reply messages. Yet, this approach provides only reliable results in settings with a large *leaf set* and a high number of replicas as shown by the results of our evaluation. Whenever either of those two parameters is reduced, as requested by the requirements of our scenario, the efficiency of this mechanism degrades.

Therefore, we proposed the *Adapted Routing Failure Test* to overcome at least a part of the restrictions of Castro et al.'s approach. Thus, replicas are no longer required for this adapted approach. Yet, this comes at the cost of degraded results in scenarios with a large *leaf set*. However, our *Adapted Routing Failure Test* provides better results than the *Routing Failure Test* in settings with either a small *leaf set* or a low number of replicas.

The *Cross-Layer Validation Mechanism* has been developed for location aware MP2P systems as Clustered Pastry. By using overheard reply messages to detect malicious nodes, a very reliable detection of forged reply messages can be ensured. As a result, the *Cross-Layer Validation Mechanism* provides better results than both mechanisms, the *Routing Failure Test* and the *Adapted Routing Failure Test* when considering a setting with an average of 12.5 virtual neighbors and without replication.

As a result of this evaluation we propose to use the *Cross-Layer Validation Mechanism* in the context of our Clustered Pastry system. However, when a MP2P system is deployed that does not provide location awareness, we propose to use the *Adapted Routing Failure Test*. This validation mechanism provides at least a moderate detection of forged reply messages.

6.7 CHAPTER SUMMARY

In the last decade, multiple MP2P systems have been proposed that are able to store data in a decentralized way in a mobile environment. Yet, by now multiple security challenges have been neglected as discussed in the previous chapter. In this chapter, we proposed new security mechanisms in order to improve the network's robustness against malicious behavior of root and intermediate nodes.

In the first part of this chapter, we discussed the influence of replicas on a MP2P system in general and on Clustered Pastry in particular. We were able to identify three challenges that can be met by replicas in MP2P systems. (I) Replicas are required to ensure the availability of objects. By storing objects, robustness against data loss due to churn can be ensured. Moreover, (II) by distributing replicas equally in the deployment area, we were able to increase the robustness against data loss in settings with a low node density. Due to these replicas, the probability could be improved that at least a single copy of an object is available even when an increased fraction of the participating nodes is disconnected from the network. Moreover, (III) replicas can be used to increase the robustness against maliciously behaving root nodes (*Storage and Retrieval Attacks*).

To meet these challenges, five replication schemes have been proposed in the context of this thesis. Our first basic replication scheme ensures that objects do not get lost due to churn by using replicas stored at the virtual neighbors of the root node (I). Moreover, we introduced three replication mechanisms that provide robustness against both, network partitioning (II), and maliciously behaving root nodes (III). Those mechanisms distribute replicas equally in the deployment area and differ on the way, the replicas are allocated. ICR uses the content provider to distribute the replicas while CRA and OCRA harnesses the root node to distribute the replicated objects. Though, OCRA provides a more reliable distribution of the replicas due to redundancy that is introduced by this approach. As a result, reliable services are provided by OCRA when considering the fraction of successful lookups and the fraction of successfully allocated objects in settings with maliciously behaving nodes or a low node density. However, our fourth replication mechanism DRA has been adapted in order to benefit from the structure of our Clustered Pastry system. Thus, DRA distributes replicas more reliable than the OCRA mechanism in settings with a high number of clusters. As a result we propose to use the OCRA mechanism in settings with up to 4 clusters and DRA whenever a high number of clusters is required.

The second part of this chapter discusses mechanisms to increase the robustness against the *Incorrect Lookup Routing Attack*. Therefore, we discussed existing security mechanisms that can be harnessed to either detect or provide robustness to maliciously behaving nodes that drop or misroute received lookup requests instead of forwarding them correctly. Yet, outcomes of this chapter have shown that those mechanisms cannot be used efficiently in the context of our Clustered Pastry system. Thus, we developed a mechanism that harness overheard messages in order to ensure that request messages are routed correctly to the destination of the request. This *Overlay WatchDog* is able to detect maliciously behaving nodes and ensures that dropped requests are rerouted. As a result, the impact of the *Incorrect Lookup Routing Attack* can be strongly decreased by applying this mechanism.

At the end of this chapter, we discussed validation mechanisms, which can be used in order to detect forged reply messages. Most of the mechanisms that have been proposed in the recent years require a centralized entity or a large *leaf set* and a set of replicas stored at virtual neighbors of the root node. The most promising approach was proposed by Castro et al. and uses the node density to detect forged reply messages. Yet, as shown in our evaluation, this *Routing Failure Test* does not provide reliable results in MP2P scenarios. As a result, we have developed an *Adapted Routing Failure Test* to overcome the drawbacks of the *Routing Failure Test*. This adapted validation mechanisms was able to provide better results in MP2P scenarios. Yet, this approach still requires equally distributed overlay identifiers and is not usable in location aware MP2P systems as Clustered Pastry. Therefore, we developed the *Cross-Layer Validation Mechanism*, that is adapted to the structure of Clustered Pastry. This mechanism harnesses the clustering of virtual neighbors and overheard messages in order to validate reply messages. As a result, the *Cross-Layer Validation Mechanism* is able to detect forged reply messages with a very high probability.

In summary, we proposed security mechanisms that can be used to match the security challenges that have been introduced in the previous chapter. Moreover, we were able to show that existing security mechanisms cannot be used in the context of MP2P scenarios or, at least, are not as efficient as our adapted mechanisms.

7

CONCLUSIONS

»I seldom end up were I wanted to go,
but almost always end up where I need to be.«

— Douglas Adams

MP2P systems can be used to provide reliable and, decentralized services in a mobile network. As a result, MP2P systems can be deployed efficiently in a disaster relief scenario. Yet, they also introduce new scientific challenges that have to be considered during the development of an MP2P system. This includes the dynamic topology and the wireless channel of the underlay and the decentralized structure of the overlay. Moreover, security challenges have to be considered as these MP2P systems are vulnerable to multiple attacks.

7.1 SUMMARY AND CONCLUSIONS

The major objective of this thesis was to develop an MP2P system that harnesses state-of-the-art technology to provide efficient and reliable storage and retrieval services in a disaster relief scenario. Therefore, our first goal was to develop a decentralized MP2P system that meet the challenges introduced by the disaster relief scenario. Based on this MP2P system, security threats had to be analyzed as our second goal. Thus, open security challenges were identified. In order to meet those open security challenges, security mechanisms had to be developed to ensure the availability of the storage and retrieval services provided by our MP2P system as our last goal.

As our first contribution and to satisfy the first goal of this thesis, we developed the Clustered Pastry system. This decentralized MP2P system provides storage and retrieval operations of small data items in a disaster relief scenario where no communication infrastructure is available. We deployed a combination of a MANET underlay with a DHT overlay in order to meet challenges that arose in a disaster relief scenario. These challenges include a dynamic topology due to the mobility of the participants, a decentralized characteristic of the network and a strongly limited bandwidth. Therefore, we harnessed the geographical position of the participants to cluster nodes in order to optimize the efficiency of the routing mechanism. Moreover, a reduced number of routing table entries was used to minimize the traffic overhead due to routing table updates. In order to ensure a reliable operation of the overlay's lookup mechanism, periodical updates of the routing tables have further been introduced. As shown in our evaluation, these adaptations were essential in order to ensure the availability of the network's services.

Furthermore, we have introduced mechanisms to optimize our Clustered Pastry system. This includes on one hand, mechanisms that reduce the traffic introduced by the update mechanisms of the routing tables. Due to those traffic reduction mechanisms, the overall traffic has been reduced by approximately 60%. On the other hand, by harnessing threshold areas, we were able to stabilize the system and to

reduce traffic generated by a node that changes the cluster. Those threshold areas are located between the clusters and delay the adaptation of the overlay identifier of a node due to its mobility pattern. We were able to show that those threshold areas reduce the average number of cluster changes by more than 10% and, therefore, the traffic generated by this cluster-based churn.

Our second contribution considers the vulnerability of MP2P systems in general. Therefore, we analyzed security threats for MP2P due to maliciously behaving participants and, in particular, routing attacks as required by our second goal. We discussed security challenges in the underlying architectures. After taking existing security mechanisms into account, we identified open challenges in MP2P. Therefore, three attacks that are able to strongly affect the availability of the services provided by the MP2P system have been analyzed in depth. Those attacks exploit the required cooperation of the participating nodes in MP2P systems. Thus, nodes do not provide locally stored objects but deny them due to the *Storage and Retrieval Attack*. Moreover, lookup requests are not forwarded by the intermediate nodes but dropped (*Incorrect Lookup Routing Attack*) or a forged reply message is generated (*Forging of Reply Messages*). In the context of this thesis, we evaluated the impact of those attacks by the means of simulations based on our Clustered Pastry simulation model. As a result, we were able to show that these attacks are also able to threaten the availability of services of MP2P systems.

In our third contribution, we proposed novel security mechanisms for MP2P system based on the results of the survey of security threats. In order to satisfy our last goal, we surveyed and evaluated existing security mechanisms that have been developed for the underlying architectures. However, as those mechanisms were not adapted to the challenges introduced by the MP2P system, these mechanisms either perform poorly or introduce a high overhead in MP2P scenarios. As a result, we developed three security mechanisms that were adapted to the characteristics of MP2P and, in particular, to Clustered Pastry systems to ensure robustness to the previously mentioned attacks.

In order to ensure the availability of the objects stored in the networks object replication is required. Therefore, we introduced replication mechanisms that are based on distributing replicas not only in the virtual overlay identifier space, but also in the deployment area. Thus, those replicas do not only ensure the robustness against the *Storage and Retrieval Attack*, but further are able reduce the impact of a network partitioning as shown in our evaluation. Moreover, we introduced replica allocation mechanisms that were able to distribute the replicas efficiently in an MP2P system. As a result of replication, the fraction of failed lookups has been reduced by nearly 35%.

During the lookup of an object, intermediate nodes are required to forward the lookup request towards its destination. However, when those intermediate nodes do not forward but drop the requests, services provided by the MP2P system may become unavailable. Existing security mechanisms that have been proposed for P2P systems to increase the networks robustness to this *Incorrect Lookup Routing Attack* are based on introducing redundancy. As the bandwidth in MP2P systems is strongly limited due to the wireless underlay, those mechanisms should be avoided. Thus, we introduced a new approach that is based on overhearing received messages and cross-layering, which provides robust services but without introducing a high overhead as

shown in our evaluation. Moreover, the impact of the *Incorrect Lookup Routing Attack* in terms of fraction of failed lookups has been reduced by more than 28%.

When an intermediate node does not drop a received lookup message but forge a reply message, new challenges arise. As the source node assumes that the lookup has been successful, security mechanisms developed for the *Incorrect Lookup Routing Attack* cannot be deployed. However, existing security mechanisms developed for P2P systems are based on requirements that cannot be met in MP2P scenarios. Therefore, we introduced an adapted mechanism that harnesses the geographical neighbors to validate reply messages. This mechanism ensures that forged reply message can be detected in a reliable way as shown in the evaluation.

In conclusion, we have developed Clustered Pastry, a location-aware and efficient MP2P architecture. In order to ensure reliable services even in settings with faulty or malicious nodes, we identified open challenges in MP2P systems and developed adapted security mechanisms to provide robustness to those attacks.

7.2 OUTLOOK

In this thesis, an MP2P system has been developed and optimized in order to be used in the context with a disaster relief scenario. Moreover, security challenges introduced by those MP2P systems have been discussed. Naturally, new research topics for future work arise based on the outcomes of this thesis.

- Clustered Pastry has been designed in the context of a disaster relief scenario. Yet, MP2P can be deployed in a large set of scenarios including vehicular networks and military operations. Thus, our system can be analyzed and optimized for other application scenarios as disaster relief.
- In this thesis we assume a fixed number of nodes and clusters. Yet, the number of participants in a disaster relief scenario may vary and, therefore, a dynamic cluster size may be beneficial to coordinate these scenarios. Thus, an algorithm that provides a dynamic number of clusters as a function of the number of participants would be an interesting topic for future work.
- In this thesis, the vulnerabilities of MP2P systems have been discussed. As a result, we considered that the network's robustness to multiple attacks is covered by existing security mechanism. For example, multiple security mechanisms have been proposed to detect a Sybil attack. The evaluation of those security mechanisms and, furthermore, optimization of these systems in the context of MP2P systems is also a topic for future work.

BIBLIOGRAPHY

[1] Welcome in the Freifunk Wiki. URL `http://wiki.freifunk.net/Kategorie:English`.

[2] ISO/IEC 7498: Information technology - Open System Interconnection - Basic Reference Model: The Basic Model, 1994.

[3] IEEE 802.11-2012: IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2012.

[4] *INET Framework for OMNeT++*, 2012. URL `http://inet.omnetpp.org/doc/INET/inet-manual-draft.pdf`.

[5] W. Acosta and S. Chandra. Trace Driven Analysis of the Long Term Evolution of Gnutella Peer-to-Peer Traffic. In S. Uhlig, K. Papagiannaki, and O. Bonaventure, editors, *Passive and Active Network Measurement*, volume 4427 of *Lecture Notes in Computer Science*, pages 42 – 51. Springer Berlin Heidelberg, 2007.

[6] M. S. Artigas, P. G. Lòpez, and A. F. G. Skarmeta. A Novel Methodology for Constructing Secure Multipath Overlays. *IEEE Internet Computing*, 9(6):50–57, 2005.

[7] N. Aschenbruck, M. Frank, P. Martini, and J. Tölle. Human Mobility in MANET Disaster Area Simulation - A Realistic Approach. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Network*, 2004.

[8] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn. BonnMotion: a mobility scenario generation and analysis tool. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, 2010.

[9] E. Auf der Heide. *Disaster Response: Principles of Preparation and Coordination*. Mosby, 2000.

[10] D. Bauer, P. Hurley, and M. Waldvogel. Replica Placement and Location using Distributed Hash Tables. In *Proceedings of the 32nd IEEE Conference on Local Computer Networks*, 2007.

[11] I. Baumgart, B. Heep, and S. Krause. OverSim: A Flexible Overlay Network Simulation Framework. In *Proceedings of the 10th IEEE Global Internet Symposium*, 2007.

[12] P. Bellavista, A. Corradi, and E. Magistretti. REDMAN: a Decentralized Middleware Solution for Cooperative Replication in Dense MANETs. In *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications Workshops*, 2005.

[13] P. Bellavista, A. Corradi, and E. Magistretti. Comparing and evaluating lightweight solutions for replica dissemination and retrieval in dense MANETs. In *Proceedings of the 10th IEEE Symposium on Computers and Communications*, 2005.

[14] *Specification of the Bluetooth System - Master Table of Contents & Compliance Requierements*. Bluetooth SIG, Inc, version 4.0 edition, 2010.

[15] R. Braden. RFC 1122: Requirements for Internet Hosts - Communication Layers, 1989.

[16] S. Buchegger, C. Tissieres, and J. Y. Le Boudèc. A Test-Bed for Misbehavior Detection in Mobile Ad-hoc Networks - How Much Can Watchdogs Really Do? In *Proceedings of the 6th IEEE Workshop on Mobile Computing Systems and Applications*, 2003.

[17] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. Wallach. Secure routing for structured peer-to-peer overlay networks. In *Proceedings of the 5th Symposium on Operating Systems Design and Implementation*, 2002.

[18] M. Castro, E. Villanueva, I. Ruiz, S. Sargento, and A. J. Kassler. Performance Evaluation of Structured P2P over Wireless Multi-hop Networks. In *Proceedings of the 2nd International Conference on Sensor Technologies and Applications*, 2008.

[19] T. C. Chan, J. Killeen, and W. Griswold. Information Technology and Emergency Medical Care during Disasters. *Academic Emergency Medicine*, 11(11):1229–1236, 2004.

[20] I. Clarke, O. Sandberg, B. Wiley, and T. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Proceedings of the ICSI Workshop on Design Issues in Anonymity and Unobservability*, 2000.

[21] T. Clausen and P. Jacquet. RFC 3626: Optimized Link State Routing Protocol (OLSR), 2003.

[22] Commissie Onderzoek Vuurwerkramp. *De vuurwerkramp Eindrapport*. Ministerie van Binnenlandse Zaken en Koninkrijksrelaties, 2002.

[23] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. A High-Throughput Path Metric for Multi-Hop Wireless Routing. *Wireless Networks*, 11(4):419–434, 2005.

[24] J. Douceur. The Sybil Attack. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems*, 2002.

[25] C. Eckert. *IT-Sicherheit Konzepte - Verfahren - Protokolle*. Oldenbourg Wissenschaftsverlag, 2005.

[26] P. Fenkam, S. Dustdar, E. Kirda, G. Reif, and H. Gall. Towards an access control system for mobile peer-to-peer collaborative environments. In *Proceedings of the 11th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2002.

[27] M. Frassl, M. Lichtenstern, M. Khider, and M. Angermann. Developing a System for Information Management in Disaster Relief - Methodology and Requirements. In *Proceedings of the 7th International Information Systems for Crisis Response and Management Conference*, 2010.

[28] R. Friedman, D. Gavidia, L. Rodrigues, A. C. Viana, and S. Voulgaris. Gossiping on MANETs: the Beauty and the Beast. *ACM SIGOPS Operating Systems Review*, 41(5):67–74, 2007.

[29] T. Fuhrmann. Performance of scalable source routing in hybrid MANETs. In *Proceedings of the 4th Annual Conference on Wireless on Demand Network Systems and Services*, 2007.

[30] L. Ganesh and B. Y. Zhao. Identity Theft Protection in Structured Overlays. In *Proceedings of the 1st IEEE ICNP Workshop on Secure Network Protocols*, 2005.

[31] N. Garg and R. P. Mahapatra. MANET Security Issues. *International Journal of Computer Science and Network Security*, 9(8):241–246, 2009.

[32] A. Ghodsi, L. Alima, and S. Haridi. Symmetric replication for structured peer-to-peer systems. In *Proceedings of the International Conference on Databases, Information Systems, and Peer-to-Peer Computing*, 2005.

[33] D. Goldschlag, M. Reed, and P. Syverson. Hiding Routing information. In R. Anderson, editor, *Information Hiding*, volume 1174 of *Lecture Notes in Computer Science*, pages 137 – 150. Springer Berlin / Heidelberg, 1996.

[34] A. Gopalan and T. Znati. PeerNet: a peer-to-peer framework for service and application deployment in MANETs. In *Proceedings of the 1st International Symposium on Wireless Pervasive Computing*, 2006.

[35] C Gottron, A König, and R Steinmetz. A Testbed-based Analysis of the Incorrect Lookup Routing Attack on the Pastry DHT. In *Proceedings of the 10th Würzburg Workshop on IP: Joint ITG, ITC, and Euro-NF Workshop V̈isions of Future Generation Networks¨*, 2010.

[36] C. Gottron, A. König, and R. Steinmetz. A Cross-Layer Approach for Increasing Robustness of Mobile Peer-to-Peer Networks. In *Proceedings of the Security in NGNs and the Future Internet Workshop*, 2010.

[37] C. Gottron, A. König, and R. Steinmetz. A Survey on Security in Mobile Peer-to-Peer Architectures Overlay-Based vs. Underlay-Based Approaches. *Future Internet*, 2(4):505–532, 2010.

[38] C. Gottron, P. Larbig, A. König, M. Hollick, and R. Steinmetz. The Rise and Fall of the AODV Protocol: A Testbed Study on Practical Routing Attacks. In *Proceedings of the 35th IEEE Conference on Local Computer Networks*, 2010.

[39] C. Gottron, A. König, and R. Steinmetz. A Cluster-Based Locality-Aware Mobile Peer-to-Peer Architecture. In *Proceedings of the 8th International Workshop on Mobile Peer-to-Peer Computing*, 2012.

[40] C. Gottron, A. König, and R. Steinmetz. Validierung von Antworten auf Objektanfragen in Mobilen Peer-to-Peer Architekturen. In *Proceedings of the 6th Workshop N̈eue Herausforderungen in der Netzsicherheit¨*, 2012.

[41] Z. Haas, M. Pearlman, and P. Samar. RFC 2026: The Zone Routing Protocol (ZRP) for Ad Hoc Networks, 2002.

[42] Z. J. Haas, J. Y. Halpern, and L. Li. Gossip-Based Ad Hoc Routing. *IEEE/ACM Transactions on Networking*, 14(3):479–491, 2006.

[43] J. Han and Y. Liu. Mutual Anonymity for Mobile P2P Systems. *IEEE Transactions on Parallel and Distributed Systems*, 19(8):1009–1019, 2008.

[44] T. Hara. Effective replica allocation in ad hoc networks for improving data accessibility. In *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, 2001.

[45] T. Hara. Replica Allocation in Ad Hoc Networks with Periodic Data Update. In *Proceedings of the 3rd International Conference on Mobile Data Management*, 2002.

[46] T. Hara. Replica Allocation Methods in Ad Hoc Networks with Data Update. *Mobile Networks and Applications*, 8(4):343–354, 2003.

[47] T. Hara, N. Murakami, and S. Nishio. Replica Allocation for Correlated Data Items in Ad Hoc Sensor Networks. *ACM SIGMOD Record*, 33(1):38–43, 2004.

[48] C. Harvesf and D. M. Blough. The Effect of Replica Placement on Routing Robustness in Distributed Hash Tables. In *Proceedings of the 6th IEEE International Conference on Peer-to-Peer Computing*, 2006.

[49] C. Harvesf and D. M. Blough. Replica Placement for Route Diversity in Tree-Based Routing Distributed Hash Tables. *IEEE Transactions on Dependable and Secure Computing*, 8(3):419 – 433, 2011.

[50] H. Hellbrück. Ad-Hoc Network Simulation, 12 2012. URL `http://www.ansim.info/`.

[51] K. Hildrum and J. Kubiatowicz. Asymptotically Efficient Approaches to Fault-Tolerance in Peer-to-Peer Networks. In *Proceedings of the 17th International Symposium on Distributed Computing*, 2003.

[52] Y. Hu, S. Das, and H. Pucha. Exploiting the Synergy between Peer-to-Peer and Mobile Ad Hoc Networks. In *Proceedings of the 9th Conference on Hot Topics in Operating Systems*, 2003.

[53] Y. C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. *Wireless Networks*, 11(1-2):21–38, 2005.

[54] International Strategy for Disaster Reduction. UNISDR Terminology on Disaster Risk Reduction, 2009.

[55] Z. Jing, W. Yijie, L. Xicheng, and Y. Kan. A dynamic adaptive replica allocation algorithm in mobile ad hoc networks. In *Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops*, 2004.

[56] D. Johnson, Y. Hu, and D. Maltz. RFC:4728 The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4, 2007.

[57] K. Kanchanasut, A. Tunpan, M. A. Awal, D. K. Das, T. Wongsaardsakul, and Y. Tsuchimoto. A Multimedia Communication System for Collaborative Emergency Response Operation in Disaster-affected Areas. Technical report, Internet Education and Research Laboratory (intERLab), Asian Institute of Technology, 2007.

[58] A. Kapadia and N. Triandopoulos. Halo: High-Assurance Locate for Distributed Hash Tables. In *Proceedings of the 15th Annual Network and Distributed System Security Symposium*, 2008.

[59] Kazaa. Download KaZaA, 2010. www.kazaa.com/.

[60] T. Kean, L. Hamilton, R. Ben-Veniste, B. Kerry, F. Fielding, J. Lehman, J. Gorelick, T. Roemer, S. Gorton, and J. Thompson. *The 9/11 Commission Report*. National Commission on Terrorist Attack, 2004.

[61] P. Kirk. Gnutella - A Protocol for a Revolution, 2010. rfc-gnutella.sourceforge.net/.

[62] P. Knežević, A. Wombacher, and T. Risse. Enabling High Data Availability in a DHT. In *Proceedings 6th International Workshop on Database and Expert Systems Applications*, 2005.

[63] A. König, M. Hollick, T. Krop, and R. Steinmetz. GeoSec: quarantine zones for mobile ad hoc networks. *Security and Communication Networks*, 2(3):271–288, 2008.

[64] A. König, M. Hollick, and R. Steinmetz. A Stochastic Analysis of Secure Joint Decision Processes in Peer-to-Peer Systems. In *Proceedings of the 3rd IEEE International Conference on Communications*, 2009.

[65] A. Kovačević. *Peer-to-Peer Location-Based Search: Engineering a Novel Peer-to-Peer Overlay Network*. PhD thesis, Technischen Universität Darmstadt, 2009.

[66] K. C. Lee, S.-H. Lee, R. Cheung, U. Lee, and M. Gerla. First Experience with CarTorrent in a Real Vehicular Ad Hoc Network Testbed. In *Proceedings of the Workshop on Mobile Networking for Vehicular Environments*, 2007.

[67] Z. Li, X. Xu, L. Shi, J. Liu, and C. Liang. Authentication in Peer-to-Peer Network: Survey and Research Directions. In *Proceedings of the 3rd International Conference on Network and System Security*, 2009.

[68] Z. Liu, A. W. Joy, and R. A. Thompson. A Dynamic Trust Model for Mobile Ad Hoc Networks. In *Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems*, 2004.

[69] H. Lu and M. K. Denko. Replica update strategies in mobile ad hoc networks. In *Proceedings of the 2nd IFIP International Conference on Wireless and Optical Communications Networks*, 2005.

[70] H. Lundgren, E. Nordström, and C. Tschudin. The Gray Zone Problem in IEEE 802.11b based Ad hoc Networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(3):104–105, 2002.

[71] M. Manulis. Privacy-preserving admission to mobile peer-to-peer groups. In *Proceedings of the 8th IEEE International Conference on Pervasive Computing and Communications Workshops*, 2010.

[72] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th International Conference on Mobile Computing and Networking*, 2000.

[73] S. Marti, P. Ganesan, and H. Garcia-Molina. DHT Routing Using Social Links. *Peer-to-Peer Systems III*, 5(1):1–6, 2004. URL `http://www.springerlink.com/index/du1cycg9yr0c7y0u.pdf`.

[74] G. Millar, T. Ramrekha, and C. Politis. A Peer-to-Peer Overlay Approach for Emergency Mobile Ad Hoc Network Based Multimedia Communications. In *Proceedings of the 5th International ICST Mobile Multimedia Communications Conference*, 2009.

[75] A. Mondal, S. Madria, and M. Kitsuregawa. CLEAR: An Efficient Context and Location-Based Dynamic Replication Scheme for Mobile-P2P Networks. In *Proceedings of the International Conference on Database and Expert Systems Applications*, 2006.

[76] A. Mondal, S. Madria, and M. Kitsuregawa. CADRE: A Collaborative replica allocation and deallocation approach for Mobile-P2P networks. In *Proceedings of the 10th International Database Engineering and Applications Symposium*, 2006.

[77] K. Needels and M. Kwon. Secure routing in peer-to-peer distributed hash tables. In *Proceedings of the ACM Symposium on Applied Computing*, 2009.

[78] J. Newsome, E. Shi, D. Song, and A. Perrig. The Sybil Attack in Sensor Networks: Analysis & Defenses. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, 2004.

[79] P. Padmanabhan, L. Gruenwald, A. Vallur, and M. Atiquzzaman. A survey of data replication techniques for mobile ad hoc network databases. *The VLDB Journal*, 17:1143–1164, 2008. doi: 10.1007/s00778-007-0055-0.

[80] C. Perkins, E. Belding-Royer, and S. Das. RFC 3561: Ad hoc On-Demand Distance Vector (AODV) Routing, 2003.

[81] C. Piro, C. Shields, and B. Levine. Detecting the Sybil Attack in Mobile Ad hoc Networks. In *Proceedings of the 2nd International Conference on Security and Privacy in Communication Networks*, 2006.

[82] J. Postel. RFC 768: User Datagram Protocol, 1980.

[83] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips. The Bittorrent P2P File-Sharing System: Measurements and Analysis. In M. Castro and R. Renesse, editors, *Peer-to-Peer Systems IV*, volume 3640 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2005.

[84] H. Pucha, S. Das, and Y. Hu. Ekta: An Efficient DHT Substrate for Distributed Applications in Mobile Ad Hoc Networks. In *Proceedings of the 6 th IEEE Workshop on Mobile Computing Systems and Applications*, 2004.

[85] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A Scalable Content-Addressable Network. In *Proceedings of the SIGCOMM*, 2001.

[86] R. Rodrigues and B. Liskov. High Availability in DHTs: Erasure Coding vs. Replication. In Miguel Castro and Robbert Renesse, editors, *Peer-to-Peer Systems IV*, volume 3640 of *Lecture Notes in Computer Science*, pages 226–239. Springer Berlin Heidelberg, 2005.

[87] B. Roh, O. Kwon, S. Hong, and J. Kim. The Exclusion of Malicious Routing Peers in Structured P2P Systems. In S. Joseph, Z. Despotovic, G. Moro, and S. Bergamaschi, editors, *Agents and Peer-to-Peer Computing*, volume 4461 of *Lecture Notes in Computer Science*, pages 43 – 50. Springer Berlin / Heidelberg, 2008. ISBN 978-3-540-79704-3.

[88] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms*, 2001.

[89] A. Rowstron and P. Druschel. Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility. In *Proceedings of the ACM Symposium on Operating Systems Principles*, 2001.

[90] M. Sànchez-Artigas, P. García Lòpez, and A. F. Skarmeta. Making Replication Secure over Structured P2P Systems: Defending against Omission Attacks. In *Proceedings of the 5th International Workshop on Databases, Information Systems, and Peer-to-Peer Computing*, 2007.

[91] M. Sànchez-Artigas, P. García Lòpez, and A. F. Skarmeta. Bypass: Providing Secure DHT Routing through Bypassing Malicious Peers. In *Proceedings of the IEEE Symposium on Computers and Communications*, 2008.

[92] M. Sànchez-Artigas, P. García Lòpez, and A. F. Skarmeta. Secure Forwarding in DHTs - Is Redundancy the Key to Robustness? In Emilio Luque, TomÃ s Margalef, and Domingo BenÃtez, editors, *Parallel Processing*, volume 5168 of *Lecture Notes in Computer Science*, pages 611–621. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-85450-0.

[93] K. Sanzgiri, B. Dahill, B. Levine, C. Shields, and E. Belding-Royer. A secure routing protocol for ad hoc networks. In *Proceedings of the 10th IEEE International Conference on Network Protocols*, 2002.

[94] B. Schneier. *Secret and Lies: Digital Security in a Networked World*. John Wiley & Sons, 2000.

[95] D. Seither, A. König, and M. Hollick. Routing performance of Wireless Mesh Networks: A practical evaluation of BATMAN advanced. In *Proceedings of the IEEE 36th Conference on Local Computer Networks*, 2011.

[96] P. Serrano, M. Zink, and J. Kurose. Assessing the fidelity of COTS 802.11 sniffers. In *Proceedings of the 28th IEEE International Conference on Computer Communications*, 2009.

[97] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[98] A. Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of the Identity-Based Cryptosystems and Signature Schemes*, 1984.

[99] E. Sit and R. Morris. Security Considerations for Peer-to-Peer Distributed Hash Tables. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems*, 2002.

[100] G. Spanoudakis, C Kloukinas, and K Androutsopoulos. Architecting secure mobile P2P systems. In *Proceedings of the 3rd International Conference on Internet Monitoring and Protection*, 2008.

[101] M. Srivatsa and L. Liu. Vulnerabilities and Security Threats in Structured Overlay Networks: A Quantitative Analysis. In *Proceedings of the 20th Annual Computer Security Applications Conference*, 2004.

[102] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2001.

[103] I. Stoica, K. Wehrle, R. Steinmetz, J. Eberspächer, R. Schollmeier, D. Schoder, K. Fischbach, C. Schmitt, V. Darlagiannis, K. Lehmann, M. Kaufmann, S. Rieche, S. Götz, H. Niedermayer, K. Aberer, A. Datta, M. Hauswirth, M. May, K. Katrinis, A. Mislove, A. Haeberlen, A. Post, P. Druschel, A. Mauthe, O. Heckmann, P. Müller, M. Hillenbrand, H. de Meer, C. Koppen, B. Stiller, J. Mischke, D. Raz, W. Nejdl, W. Siberski, W. Balke, G. Hasslinger, K. Tutschku, P. Tran-Gia, W. Kellerer, A. Heinemann, M. Müllhäuser, O. Waldhorst, C. Lindemann, J. Kangasharju, T. Hummerl, S. Muhle, J. Gerke, D. Hausheer, M. Conrad, H.Hartenstein, M. Schüller, M. Zitterbart, D. Rolli, R. Ackermann, L. Divic-Krnic, N. Liebau, and T. Roscoe. *Peer-to-Peer Systems and Applications*. Springer-Verlag Berlin Heidelberg, 2005.

[104] D. Stutzbach and R. Rejaie. Understanding churn in peer-to-peer networks. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet measurement*, 2006.

[105] N. Suri, G. Benincasa, M. Tortonesi, C. Stefanelli, J. Kovach, R. Winkler, R. Kohler, J. Hanna, L. Pochet, and S. Watson. Peer-to-Peer Communications for Tactical Environments: Observations, Requirements, and Experiences. *IEEE Communications Magazine*, 48(10):60–69, 2010.

[106] M. Tamori, S. Ishihara, T. Watanabe, and T. Mizuno. A replica distribution method with consideration of the positions of mobile hosts on wireless ad-hoc networks. In *Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops*, 2002.

[107] A. Tanenbaum and D. Wetherall. *Computer Networks*. Prentice Hall Professional Technical Reference, 5th revised edition edition, 2010.

[108] K. Tsai, C. Hsu, and T. Wu. Mutual anonymity protocol with integrity protection for mobile peer-to-peer networks. *International Journal of Security and Networks*, 5(1):45–52, 2010.

[109] *UNDAC HANDBOOK*. United Nations Disaster Assessment and Coordination, 2006.

[110] *BonnMotion a Mobility Scenario Generation and Analysis Tool*. University of Bonn, 2011. URL `https://net.cs.uni-bonn.de/fileadmin/ag/martini/projekte/BonnMotion/src/BonnMotion_Docu.pdf`.

[111] A. Varga. *OMNeT++ User Manual - Version 4.2.2*. OpenSim Ltd., 2011.

[112] S. Čapkun, J. P. Hubaux, and L. Buttyán. Mobility Helps Peer-to-Peer Security. *IEEE Transactions on Mobile Computing*, 5(1):43–51, 2006.

[113] James Walkerdine, Peter Phillips, and Simon Lock. *Mobile Peer-to-Peer Computing for Next Generation Distributed Environments: Advancing Conceptual and Algorithmic Applications*, chapter A Tool Supported Methodology for Developing Secure Mobile P2P Systems, pages 283–300. IGI Global, 2009.

[114] P. Wang, I. Osipkov, N. Hopper, and Y. Kim. Myrmic : Secure and Robust DHT Routing. Technical report, University of Minnesota, 2006. URL `http://www-users.cs.umn.edu/~kyd/paper/wohk07.pdf`.

[115] D. Wu, Y. Tian, and K.-W. Ng. Analytical Study on Improving DHT Lookup Performance under Churn. In *Proccedings of the 6th IEEE International Conference on Peer-to-Peer Computing*, 2006.

[116] D. Wu, Y. Tian, K.-W. Ng, and A. Datta. Stochastic analysis of the interplay between object maintenance and churn. *Computer Communications*, 31(2):220 – 239, 2008.

[117] H. Yang, F. Ricciato, S. Lu, and L. Zhang. Securing a Wireless World. *Proceedings of the IEEE*, 94(2):442–454, 2006.

[118] T. Zahn and J. Schiller. MADPastry: A DHT Substrate for Practicably Sized MANETs. In *Proceedings of the 5th IEEE Workshop on Applications and Services in Wireless Networks*, 2005.

[119] M. Zapata and N. Asokan. Securing ad hoc routing protocols. In *Proceedings of the 1st ACM Workshop on Wireless Security*, 2002.

[120] Q. Zhang and D. Agrawal. Dynamic probabilistic broadcasting in MANETs. *Journal of Parallel and Distributed Computing*, 65(2):220 – 233, 2005.

[121] B. Zhao, L. Huang, J. Stribling, S. Rhea, A. Joseph, and J. Kubiatowicz. Tapestry: a resilient global-scale overlay for service deployment. *IEEE Journal on Selected Areas in Communications*, 22(1):41–53, 2004.

[122] M. Zink and A. Mauthe. P2P Streaming using Multiple Description Coded Video. In *Proccedings of the Euromicro Conference*, 2004.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ANSim	Ad-hoc Network Simulator	57
AODV	Ad hoc On-Demand Distance Vector	10
CAN	Content Addressable Network	13
CRA	Cyclic Replica Allocation	105
DHT	Distributed Hashtable	4
DRA	Delegate Replica Allocation	108
DSR	Dynamic Source Routing	10
ETX	Expected Transmission Count	11
GPS	Global Positioning System	2
ICR	Inter-Cluster Replication	102
IDS	Intrusion Detection System	22
IP	Internet Protocol	12
IRS	Intrusion Response System	82
ISDR	International Strategy for Disaster Reduction	28
MANET	Mobile Ad hoc Network	1
MP2P	Mobile Peer-to-Peer	2
MPR	Multi Point Relay	10
OCRA	Optimized Cyclic Replica Allocation	106
OLSR	Optimized Link State Routing protocol	10
OSI	Open Systems Interconnection	8
P2P	Peer-to-Peer	2
RFC	Request for Comments	57
RTT	Round Trip Time	96
TC	Topology Control	10
UDP	User Datagram Protocol	35
WLAN	Wireless Local Area Networks	27

A

APPENDIX

A.1 DEFAULT PARAMETERS

In this section, default parameters are defined as used by our simulation based evaluation.

A.1.1 *Field Size and Basic Parameters*

Field size	
Nodes	Deployment area
25	500m x 500m
50	800m x 800m
75	900m x 900m
100	1100m x 1100m
150	1400m x 1400m
200	1550m x 1550m
Basic parameters	
Nodes	100
Mobility model	Random waypoint
Node speed	0-1 m/s
Node placement	Equally distributed

A.1.2 *Overlay*

Pastry	
Basic protocol	Pastry
Routing algorithm	semi-recursive
Leaf set size	dynamic
ID size	32 bit
Parameter b	1
Retransmissions	2
Number of local replicas	2
Routing table update frequency	10 s
Traffic generator	
Request frequency	10 s (global)
Objects	50
Life time of Objects	250 s
Object size	2 kbyte - 10 kbyte

A.1.3 *Underlay*

Underlay	
Carrier frequency	2.4 GHz
Propagation model	FreeSpaceModel
Transmission range	200m
MANET parameters	
Basic protocol	OLSR
Routing metric	ETX
Frequency Hello messages	2 s
Willingness	3

A.1.4 *Clustered Pastry*

Clustered Pastry	
Threshold area	40m
Cluster	4
Leaf Set update frequency	2 s
Traffic reduction mechanism	
Update interval of the data sets	6s
Data set size	6 bytes

B

AUTHOR'S PUBLICATIONS

B.1 MAIN PUBLICATIONS

1. Christian Gottron, André König, and Ralf Steinmetz. *A Cluster-Based Locality-Aware Mobile Peer-to-Peer Architecture*. In: *Proceedings of the 8th International Workshop on Mobile Peer-to-Peer Computing*. 2012.

2. Christian Gottron, André König, and Ralf Steinmetz. *Validierung von Antworten auf Objektanfragen in Mobilen Peer-to-Peer Architekturen*. In: *Proceedings of the 6th Workshop N̈eue Herausforderungen in der Netzsicherheit¨*. 2012.

3. Christian Gottron, André König, and Ralf Steinmetz. *A Cross-layer Approach Towards Robustness of Mobile Peer-to-Peer Networks*. In: *Proceedings of the 7th IEEE International Workshop on Wireless and Sensor Networks Security*. 2011.

4. Christian Gottron, André König, and Ralf Steinmetz. *A Testbed-based Analysis of the Incorrect Lookup Routing Attack on the Pastry DHT*. In: *Proceedings of the 10th Würzburg Workshop on IP: Joint ITG, ITC, and Euro-NF Workshop V̈isions of Future Generation Networks¨*. 2010.

5. Christian Gottron, André König, and Ralf Steinmetz. *A Cross-Layer Approach for Increasing Robustness of Mobile Peer-to-Peer Networks*. In: *Proceedings of the Security in NGNs and the Future Internet Workshop*. 2010.

6. Christian Gottron, André König, and Ralf Steinmetz. *A Survey on Security in Mobile Peer-to-Peer Architectures Overlay-Based vs. Underlay-Based Approaches*. *Future Internet 2*, (4):505–532, 2010.

7. Christian Gottron, Pedro Larbig, André König, Matthias Hollick, and Ralf Steinmetz. *Testbed Evaluation eines Black Hole-Angriffes auf ein Ad hoc Netz*. In: *Proceedings of the 4th Workshop N̈eue Herausforderungen in der Netzsicherheit¨*. 2010.

8. Christian Gottron, Pedro Larbig, André König, Matthias Hollick, and Ralf Steinmetz. *The Rise and Fall of the AODV Protocol: A Testbed Study on Practical Routing Attacks*. In: *Proceedings of the 35th IEEE Conference on Local Computer Networks*. 2010.

9. Christian Gottron, Daniel Seither, André König, and Ralf Steinmetz. *A Testbed-based Visualization of Misbehavior in Peer-to-Peer Systems*. In: *Proceedings of the 10th Würzburg Workshop on IP: Joint ITG, ITC, and Euro-NF Workshop V̈isions of Future Generation Networks¨*. 2010.

10. Christian Gottron, André König, Matthias Hollick, Sonja Bergsträßer, Tomas Hildebrandt, and Ralf Steinmetz. *Quality of Experience of Voice Communication in Large-Scale Mobile Ad Hoc Networks*. In: *Proceedings of the 2nd IFIP Wireless Days*. 2009.

B.2 CO-AUTHORED PUBLICATIONS

11. Sebastian Fiebig, Melanie Siebenhaar, Christian Gottron, and Ralf Steinmetz. *Detecting VM Live Migration Using a Hybrid External Approach*. In: *Proceedings of the 3rd International Conference on Cloud Computing and Services Science*. 2013.

12. Christian Gross, Dominik Stingl, Christian Gottron, Björn Richerzhagen, Christoph Münker, and David Hausheer. *Harnessing Mobile Ad Hoc Communication for Decentralized Location-Based Services*. Technical Report PS-TR-2012-02, TU Darmstadt, 2012.

13. André Miede, Nedislav Nedyalkov, Christian Gottron, André König, Nicolas Repp, and Ralf Steinmetz. *A generic metamodel for it security - attack modeling for distributed systems*. In: *Proceedings of the 5th International Conference on Availability, Reliability and Security*. 2010.

14. André Miede, Christian Gottron, André König, Nedislav Nedyalkov, Nicolas Repp, and Ralf Steinmetz. *Cross-organizational security in distributed systems*. Technical Report KOM-TR-2009-01, Technische Universität Darmstadt, 2009.

15. André König, Christian Gottron, Matthias Hollick, and Ralf Steinmetz. *Harnessing delay tolerance to increase delivery ratios in mobile ad hoc networks with misbehaving nodes*. In: *Proceedings of the 4th IEEE International Workshop on Wireless and Sensor Networks Security*. 2008.

C

CURRICULUM VITÆ

PERSONAL INFORMATION

Name	Christian Gottron
Date of Birth	October 28, 1979
Place of Birth	Mainz
Nationality	German

EDUCATION

01/2009–today	Technische Universität Darmstadt (Darmstadt, Germany) PhD candidate at the Multimedia Communications Lab (KOM) Department of Electrical Engineering
10/2012–12/2012	University of Massachusetts Amherst (Amherst, USA) Visiting researcher at the department of Electrical and Computer Engineering
10/2003–11/2008	Technische Universität Darmstadt (Darmstadt, Germany) Dipl.-Ing. at the Electrical Engineering and Information Technology Department
Until 03/2003	German public high-school (Mainz, Germany)

PROFESSIONAL EXPERIENCE

01/2009–today	Technische Universität Darmstadt (Darmstadt, Germany) Research assistant at the Multimedia Communications Lab (KOM)
09/1996–01/2000	Procter and Gamble (Mainz, Germany) Job training (energy electronics technician)

TEACHING ACTIVITIES

2009–today	Communication Networks I
2009–2011	Seminar Sicherheit in mobilen Ad-hoc-Netzen (supervisor)
2009	Seminar Communication Systems and Multimedia I – Advanced Topics of Future Internet Research (supervisor)
2009–today	Tutor for various Bachelor's and Master's theses (including "Studien-" and "Diplomarbeiten")

D

ERKLÄRUNG LAUT §9 DER PROMOTIONSORDNUNG

Ich versichere hiermit, dass ich die vorliegende Dissertation allein und nur unter Verwendung der angegebenen Literatur verfasst habe.

Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Darmstadt, 2013